MARITIME TRANSPORTATION RESEARCH AND EDUCATION CENTER
TIER 1 UNIVERSITY TRANSPORTATION CENTER
U.S. DEPARTMENT OF TRANSPORTATION

The Resilience of the Port of Freeport During the Extreme Weather Events

12/01/2021 – 08/31/2023

Jim Kruse, M.S (PI)
j-kruse@tti.tamu.edu
Texas A&M Transportation Institute
Andrew Birt, Ph.D. (Co-PI)
a-birt@tti.tamu.edu
Texas A&M Transportation Institute
Rodolfo Souza, Ph.D.
r-souza@tti.tamu.edu
Texas A&M Transportation Institute

September 15, 2023

FINAL RESEARCH REPORT
Prepared for:
Maritime Transportation Research and Education Center

ACKNOWLEDGMENT

This material is based upon work supported by the U.S. Department of Transportation under Grant Award Number 69A3551747130. The work was conducted through the Maritime Transportation Research and Education Center at the University of Arkansas.

DISCLAIMER

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated in the interest of information exchange. The report is funded, partially or entirely, by a grant from the U.S. Department of Transportation's University Transportation Centers Program. However, the U.S. Government assumes no liability for the contents or use thereof.

CONTENTS

Figures

Executive Summary

The goal of this study is to evaluate data and models useful for assessing the resilience of port operations to road closures caused by storm surge flooding.

To achieve this goal, we used a case study of Port Freeport, Texas, and developed a prototype web-based application that can be used to a) predict storm surge flooding; b) predict road closures caused by flooding; c) quantify the ability of the network to maintain transport service to the port. The prototype application enabled us to integrate models and data into a tool that could be used by stakeholders to explore the relationship between flooding and transportation. However, we also attempted to identify models and data in the context of extending the prototype to other port systems, or to other applications that involve flooding, road closures, and impacts on transportation service.

Our prototype web application demonstrates that it is possible to develop software that could be used by stakeholders to explore the impact of flooding on transportation. Our initial goal was to develop an application that could be explored in real-time, such that the algorithms can be integrated to provide almost immediate results to the user. In reality, the prototype delivers results in the order of minutes rather than seconds. However, we suggest that improvements in computer hardware and the efficiency of algorithms could achieve performance that enables real-time analysis, and therefore would allow multiple stakeholders to use the application to effectively explore flood-transportation resilience.

In the process of developing the web application, we researched and explored several data sources and algorithms useful for investigating flood-transportation resilience. In particular, we consider aerial LIDAR data as a potentially important resource for researchers interested in port studies, especially those involving hydrology. Open-source aerial LIDAR data is widely available throughout the US, especially for coastal, littoral, riverine, and flood-prone urban areas. We used the aerial LIDAR data to obtain high-resolution representations of road elevations to map road closures and suggest how this data could also be used to develop models of buildings, utilities, river channels, or other structures that would be useful for assessing the impact of floods. We also used OpenStreetMap data paired with graph libraries to develop custom routing algorithms to assess the impact of road closures on transportation service to Port Freeport. We discuss extensions and modifications of the algorithms for developing routing algorithms for other niche applications.

In the context of resilience and resilience thinking, we suggest that our approach of developing and integrating open-source data and algorithms is a potentially useful way of promoting effective collaboration between hydraulic engineers, transportation specialists, port managers, and other stakeholders. Data and models already exist that are useful for exploring the interactions among hurricanes, flooding, road closures, and transportation impacts. The integration of these models and data into a practical tool provides a useful construct to explore how data, models and methods can be appropriately simplified to achieve a specified design goal.

Introduction

The goal of this project is to conceptualize and develop an information technology (IT) framework that can be used to assess the impact of flooding on the resilience of port operations. Specifically, the project is focused on evaluating the availability of data and models that can be used to help predict the impact of floods on regional transportation links, and in doing so enable stakeholders to improve the resilience of port infrastructure and operations.

According to the United States Department of Homeland Security, ninety percent of all-natural disasters involve flooding[1]. Due to their proximity to water, seaports and riverports are designed to withstand historical flood regimes and many of the dangers posed by local flooding. However, for these ports to function, they must be serviced by inland transportation infrastructure such as roads and railways, and by other service infrastructure such as electricity, fuel lines, communication cables, etc. Most often, the planning and management of this infrastructure are outside the direct control of port managers. Many of the skills and data required to predict flooding or transportation analysis may also be outside the conventional skill set of port managers. The purpose of this project is to explore the potential for IT-based approaches to be used to help predict flood events and help port managers adapt and respond to flood events.

This project uses a case study approach to explore the availability of data and algorithms to predict and manage flooding. The case study focuses on Port Freeport in Texas. Port Freeport is situated approximately 60 miles south of downtown Houston, in the city of Freeport. Port Freeport is ranked 19th in the nation for total tonnage and is one of the top 10 fastest-growing seaports in the nation[2]. Port Freeport also serves several large, geographically diverse cities and regions within Texas and beyond. The port therefore relies on a complex hinterland transportation system to maintain the flow of goods and services to and from the port.

While Port Freeport serves as a useful case study, one of the goals of this project is to evaluate the potential of data and models to assess flood resilience for any port—especially smaller ports that may not have the human resources to evaluate and respond to flood risk. Therefore, a significant objective of this report is to discuss the advantages and disadvantages of our approaches, the availability of data for other port systems, and opportunities for extending and improving the models and data presented herein.

Resilience, Ports, and Flooding

Resilience has become an increasingly important term and concept for transportation planners and practitioners. In transportation and many other fields resilience intuitively means that a system or object can bounce back or regain shape following a stressful event or disturbance; however, resilience has become a widely used and useful planning and management concept that has been progressively turned into formal definitions, policy, and law. For example, Executive Order 13693 defines resilience as:

[1] https://www.dhs.gov/natural-disasters accessed August 29th, 2023.
[2] https://www.portfreeport.com/explore accessed August 29th, 2023.

> "...the ability to anticipate, prepare for, and adapt to changing conditions and withstand, respond to, and recover rapidly from disruptions."[3]

This definition is also used by the U.S. Army Corps of Engineers[4] and the U.S. Department of Transportation Federal Highway Administration[5] among others. Touzinsky et al. (2018)[6] provide a definition that alludes to the importance of systems theory in assessing resilience:

> "…a system's ability to maintain a given critical function through preparing, resisting, recovering, and adapting to a disturbance."

This definition introduces two additional key terms—system and disturbance—that we define as follows:

- **System:** a *system* is defined in the engineering and mathematical sense as any collection of objects or relationships between objects that result in a consistent output or behavior. Figure 1 shows a conceptual view of the system explored in this project. This system is made up of a transportation network (roads) that links a port (Port Freeport in the case study) to inland areas where the port and its users need to be accessible. These include residential areas that port employees live in, shops or supply houses that are used by the port to enable maintenance and repairs, and most obviously, inland locations that provide the goods that are transported through the port. For any specific location (origin or destination) of interest, the performance of the road system can be defined by various metrics that measure the cost of travel (e.g., travel time, travel distance, or travel cost in dollars) to reach the port from that location or vice-versa. Since most ports require transportation service to multiple locations to remain functional, the overall level of service provided by the road network is a function of the total travel time for all important routes to the port. Under normal circumstances (i.e., no changes to the system) this system performance will tend towards a consistent state (equilibrium).
- **Disturbance:** A disturbance is defined as a natural or human-induced event that has the potential to impact a system (e.g., a port hinterland transportation system). Disturbances are normally thought of as events that occur independently or external to an otherwise closed system of interest (Birt and Coulson, 2015)[7]. For example, hurricanes are best modeled as occurring independently of the port hinterland transportation system because the road system does not directly generate hurricanes. Instead, it is more practical to consider hurricanes as an independent, external force that has an effect on the system but is generated by processes not explicitly included in the system of interest.

[3] https://www.govinfo.gov/content/pkg/DCPD-201500184/pdf/DCPD-201500184.pdf. Accessed August 2023.

[4] https://www.usace.army.mil/missions/sustainability/building-resilience/#:~:text=Resilience%20is%20the%20ability%20to,disruptions%20(Executive%20Order%2013563). Accessed August 2023

[5] https://highways.dot.gov/research/infrastructure/resilient-pavements/definitions. Accessed August 2023.

[6] Touzinsky, K. F., Scully, B. M., Mitchell, K. N., & Kress, M. M. (2018). Using empirical data to quantify port resilience: Hurricane Matthew and the southeastern seaboard. Journal of Waterway, Port, Coastal, and Ocean Engineering, 144(4), 05018003.

[7] Birt, Andrew G., and Robert N. Coulson. "Southern pine beetle herbivory in the southern United States: moving from external disturbance to internal process." Simulation modeling of forest landscape disturbances (2015): 165-200.

As the right-hand side of Figure 1 suggests, storm surge flooding has the potential to cause road closures that reduce the performance of the road system (e.g., travel time). Depending on the location, extent, and depth of flooding, the initial loss of service to the system may be small or large. Factors such as flood dissipation time, the extent of permanent damage, and the availability of repair crews will affect the time taken for the system to recover, or in extreme cases, govern whether the system ever recovers.

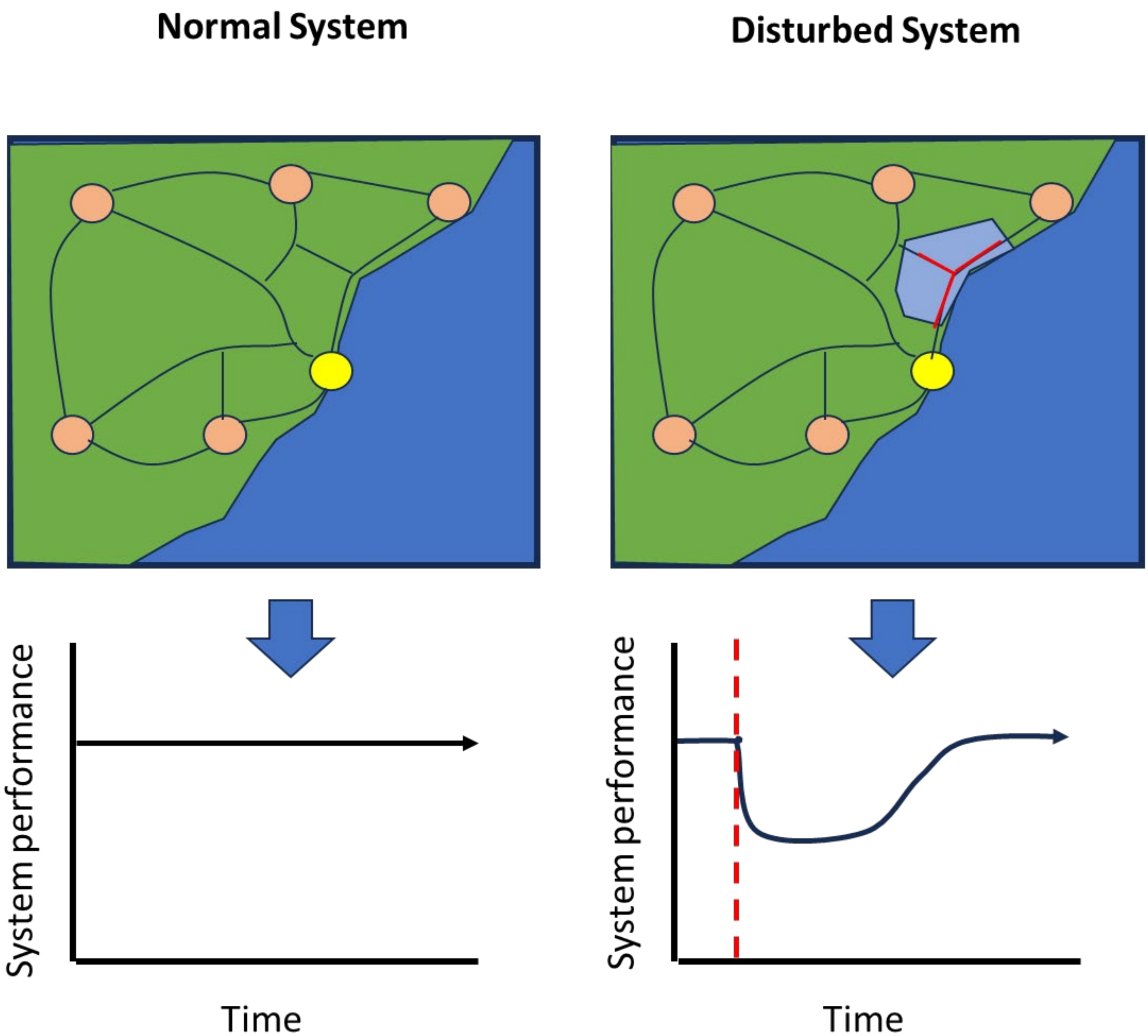

Figure 1. Conceptual diagram of a port-hinterland transportation system.
The left panel shows an undisturbed system where the cost of travel to and from the port remains constant or relatively constant through time. In the right panel, a flood (light blue area) temporarily causes temporary road closures that cause a measurable drop in system performance.

Resilience Thinking

Intuitively, definitions of resilience suggest that agents of change or disturbances are inevitable in the modern world. In many cases, it is impractical to completely prevent agents of change. It may also be impractical to engineer systems to completely withstand such disturbances. Instead, the resilience concept calls for systems to be re-engineered to absorb **and** recover from disturbances.

Resilience thinking is about approaches and concepts that can be used to attain resilient systems. The following components of resilience are useful for interpreting the methods, results, and discussions presented in this study:

1. Under normal conditions, the individual elements of systems tend towards a normal or equilibrium state. This condition is measurable and provides a useful baseline from which to measure the impact of any disturbance.
2. A disturbance has the potential to destabilize a system away from its normal operation (equilibrium). Disturbances can be short-term (pulsed) or long-term and gradual (press disturbances).
3. In quantitative systems, the impact of a disturbance on a system can be measured and compared against the baseline or normal (equilibrium) system function. Moreover, it is possible to partition the effects of a disturbance into an initial shock or outage followed by a recovery period. Figure 2 illustrates a conceptual example of four different responses to a disturbance. Condition A is the most desirable, while condition B is the least desirable. However, it is also possible that condition A is infeasible or uneconomical. Curves C and D illustrate two responses where the system is affected but in different ways.
4. Once methods to characterize disturbances and measure resilience have been identified, they can be used to adapt and redesign the system to make it more resilient. The potential for a system to be reengineered or even to self-adapt to disturbances is often referred to as adaptive capacity or adaptation.

The systems theory approach to resilience encourages the development of methods and thought processes to:

1. Understand the organization and interconnection of individual components that make up a system.
2. Identify, define, and characterize potential system disturbances.
3. Define system performance metrics and measure, model, or estimate them under equilibrium and disturbed states.
4. Redefine the boundaries of systems to include new entities and relationships that may have the potential to minimize the impact of potential disturbances either by reducing the initial impact of the disturbance, improving recovery time, or both.

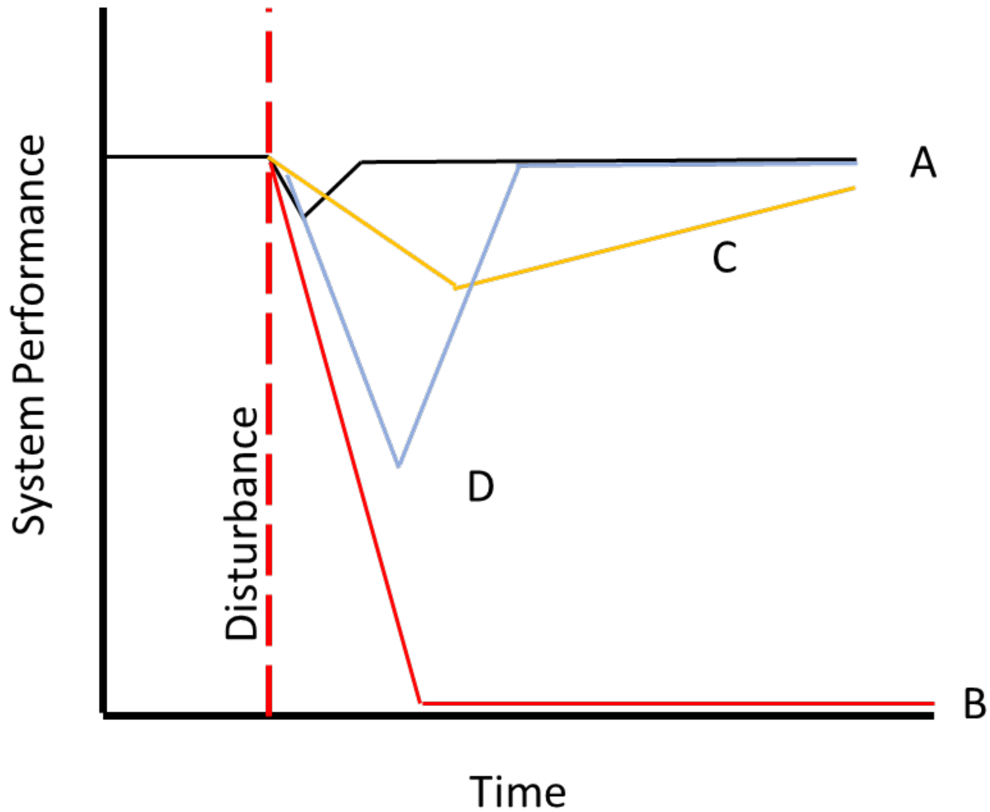

Figure 2. Four different curves that illustrate hypothetical responses to a disturbance.
Curve A suggests only a minimum loss of system performance following a disturbance and a rapid recovery. Curve B illustrates a system that is 'broken' by the disturbance and never recovers. Curve C shows a situation where the disturbance results in a relatively small drop in function but takes considerable time to recover. Curve D illustrates a system that experiences a relatively large drop in performance following a disturbance but recovers quickly.

Assessing the Resilience of Port-Hinterland Transportation to Storm Surge Flooding

This section describes a conceptual IT tool to help assess the resilience of port-hinterland transportation networks to storm surges. The tool was developed in line with the following objectives:

1. Provide quantitative metrics of resilience to storm-surge flooding.
2. Provide a quantitative, interactive resilience planning tool to help bridge knowledge gaps among stakeholders charged with improving system resilience.
3. Evaluate the availability of data and models that can be used to assess flood-transportation resilience.

The tool is developed as a web application that predicts and measures the impact(s) of hurricane storm surge flooding on the service or performance of the port-hinterland transportation system. It does this through the following linked functionality:

1. Given a specified storm surge height (from a tropical storm or hurricane), it predicts flooding (flood extent and elevation) based on the topology of the surrounding landscape and then uses this information to predict road closures or at-risk roads.
2. A user can define several origin/destination locations using the inbuilt map and run routing analyses to estimate the impact of flooding on road transportation to and from the port. The application calculates various metrics that describe the overall impact of the specified storm surge event on the port-hinterland transportation system.
3. The web application provides additional features to identify regional transportation impacts and to determine key roadway links useful for routing analyses.

The web application is focused on Port Freeport and is preloaded with data for the region. As illustrated in Figure 3 the first step in using the web application is to simulate a storm surge event by specifying a storm surge height using the slider marked in panel A. The flood extent is calculated on the server and then displayed on the built-in web map (the blue layer in panel A). The second step involves querying roads that could be disturbed by the predicted flooding—shown by the red dots in panel B. Panel C illustrates that additional flooded areas can be drawn onto the map by hand.

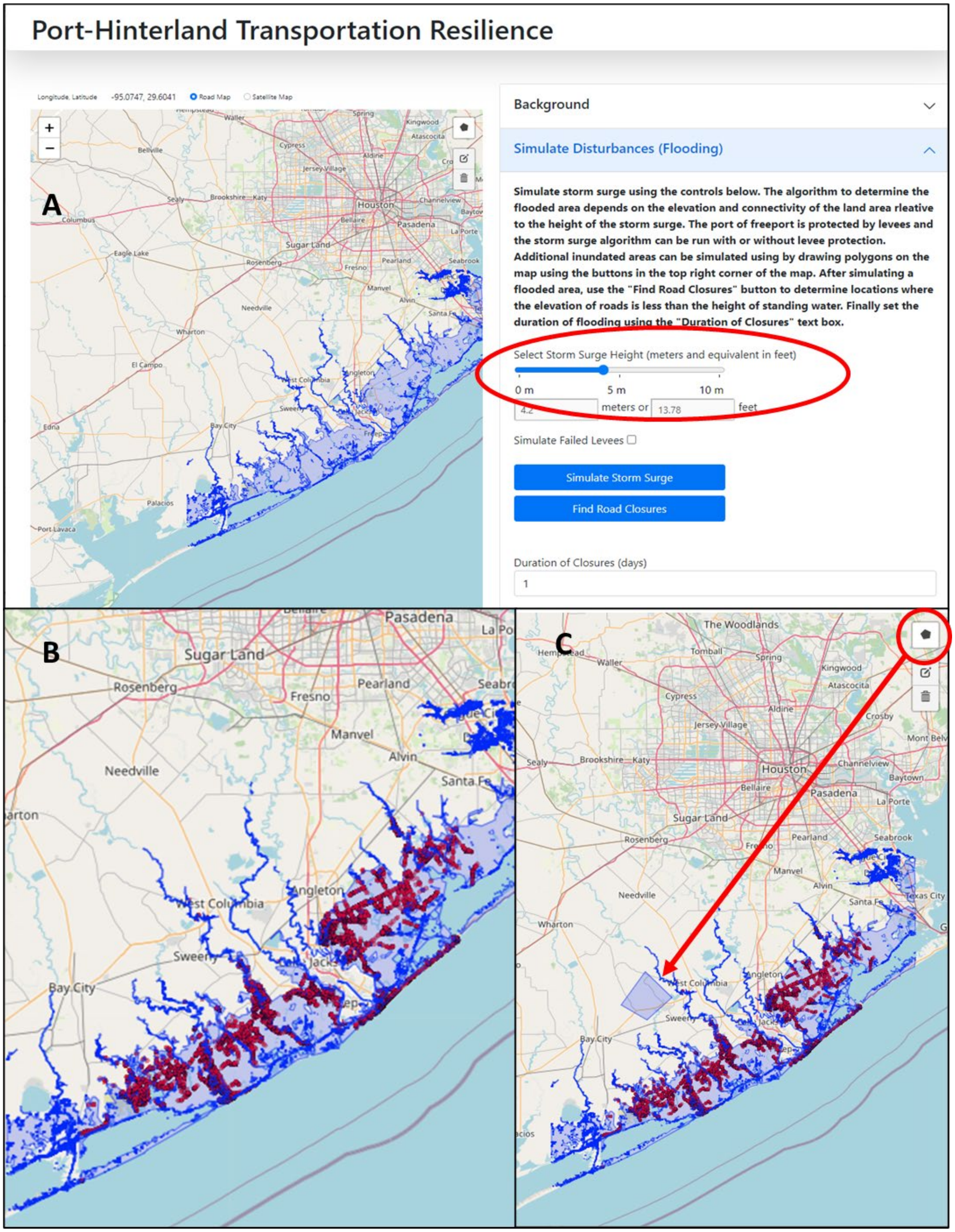

Figure 3. Screenshot of the draft web application.

After defining a disturbance (storm surge) and estimating the extent of flooding, the user can click the map to enter one or more origin/destination locations (labeled 1 and 2 in Panel A of Figure 4). Panel B in Figure 4 shows the shortest path routes to the port. The black lines show the shortest path for the undisturbed network, while the red paths show the shortest paths for the disturbed network. Finally, the difference in travel time (or the disturbance impact is calculated for the routes (Figure 4, Panel C).

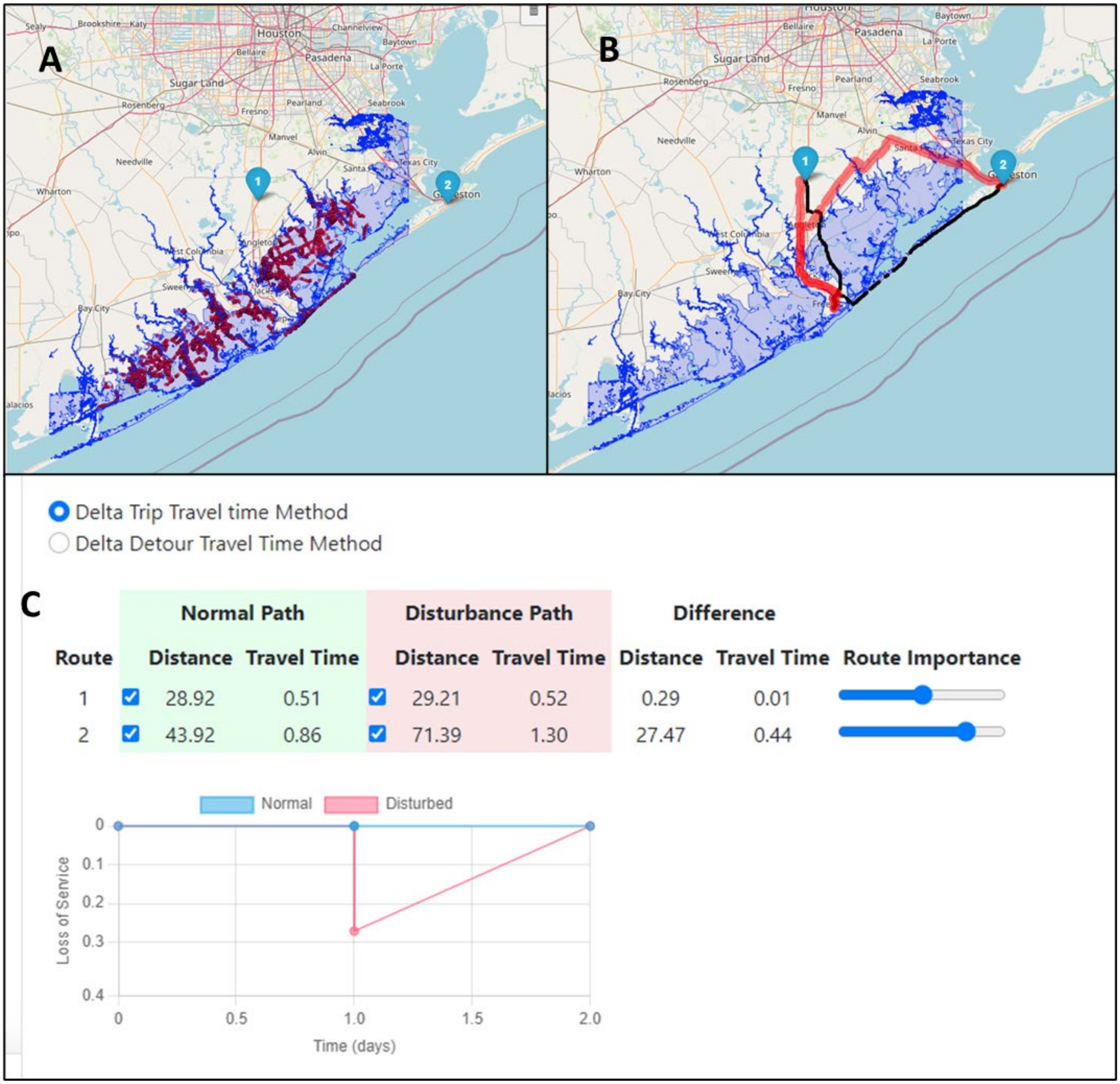

Figure 4. Screenshot of the draft web application illustrating the effect of storm surge on the road network.

The application includes other features useful for assessing port-hinterland transportation resilience. The left panel in Figure 5 shows a regional accessibility layer for a given flood, where the colored lines show the difference in travel time for an undisturbed vs. disturbed network (red areas show a large impact, yellow and orange areas a medium impact, and green areas minimal impact of a predicted flood). The right panel in Figure 5 shows the results of an algorithm that can be used to identify critical links in the road network.

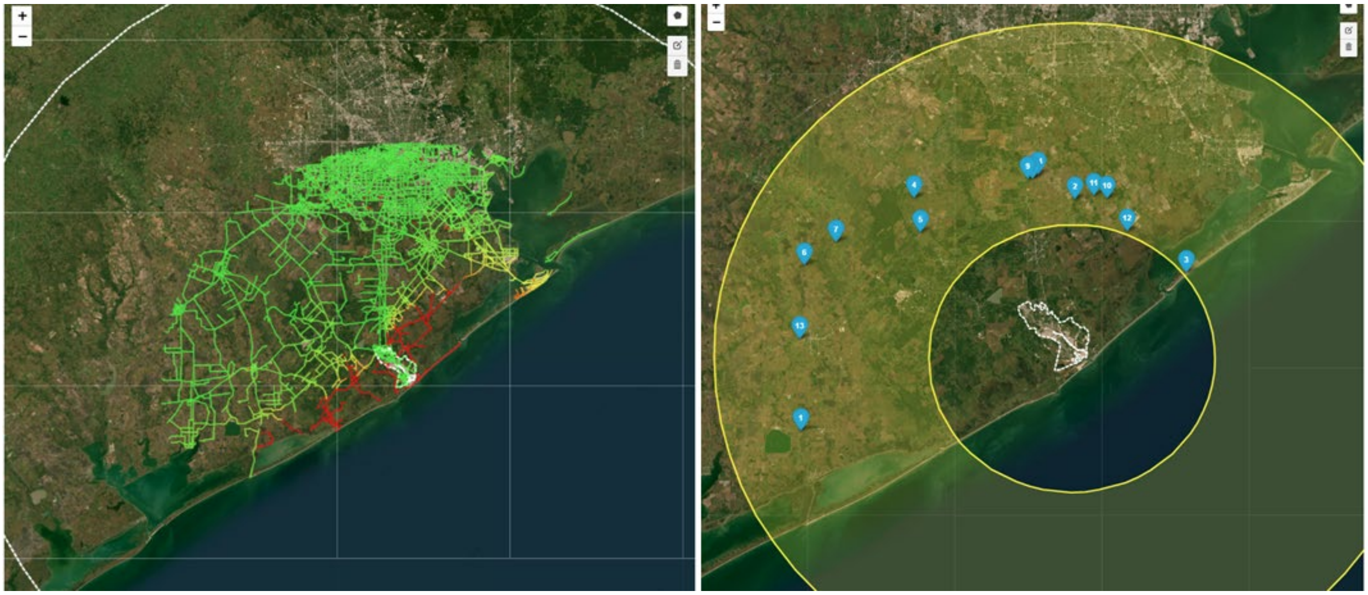

Figure 5. Screenshot of the draft web application illustrating a regional effect of disturbance and identification of critical links.

The remaining sections of this report describe the ideas, data, algorithms, and implementation of the web-based application in more detail. These descriptions should help interested readers modify or extend our ideas and methods for their purposes. To this end, as well as describing the current data and algorithms, each section will outline challenges that the research team has encountered during development and ideas for improvement or reuse of algorithms and data. Data and algorithms described in the following sections are also available at https://github.com/ttitamu/Port_resilience.

Calculating Storm Surge Flooding

The key idea behind our models of storm surge flooding is that flooding will occur as a function of the height of a storm surge as well as the local elevation of the surrounding landscape. Specifically, the idea is that flooding will occur at locations where two conditions are met:

1. An inland location has an elevation lower than the storm surge height.
2. The inland location is hydraulically connected to other locations with elevations lower than the storm surge height.

We used digital elevation models (DEM) to represent the elevation of the landscape around Port Freeport. According to the USGS, "A Digital Elevation Model (DEM) is a representation of the bare ground (bare earth) topographic surface of the Earth excluding trees, buildings, and any other surface objects."[8] .Digital elevation models are usually available in the form of Geographic Information System (GIS) raster data. Raster-based DEM models divide a region (the raster extent) into regular cells, each of which describes the mean elevation of the earth's surface at that point of location. DEMs are widely available for nearly all areas of the United States and in a variety of spatial resolutions (i.e., the size of each cell in the model) and file formats. The United States Geological Survey (USGS) provides publicly available DEMs at a variety of resolutions through its national map web application [9].

We used a Breadth First Search (BFS) to estimate flooding based on a DEM, and a specified storm surge elevation (the height of a storm surge above sea level)- Many online resources are available to explain depth-first search algorithms, so the explanation below focuses on its application to storm surge flood prediction.

The BFS algorithm is initialized by supplying the DEM and a storm surge elevation. Figure 6 shows the DEM used in the Port Freeport case study. The DEM consists of 12,559 columns and 10,525 rows of cells (132,183,475 cells in total), with each cell measuring 10 meters by 10 meters. Each cell contains the elevation of the cell (in meters) above regular sea level.

[8] https://www.usgs.gov/faqs/what-digital-elevation-model-dem#:~:text=A%20Digital%20Elevation%20Model%20(DEM)%20is%20a%20representation%20of%20the,derived%20primarily%20from%20topographic%20maps. Accessed July 2023.
[9] https://apps.nationalmap.gov/downloader/

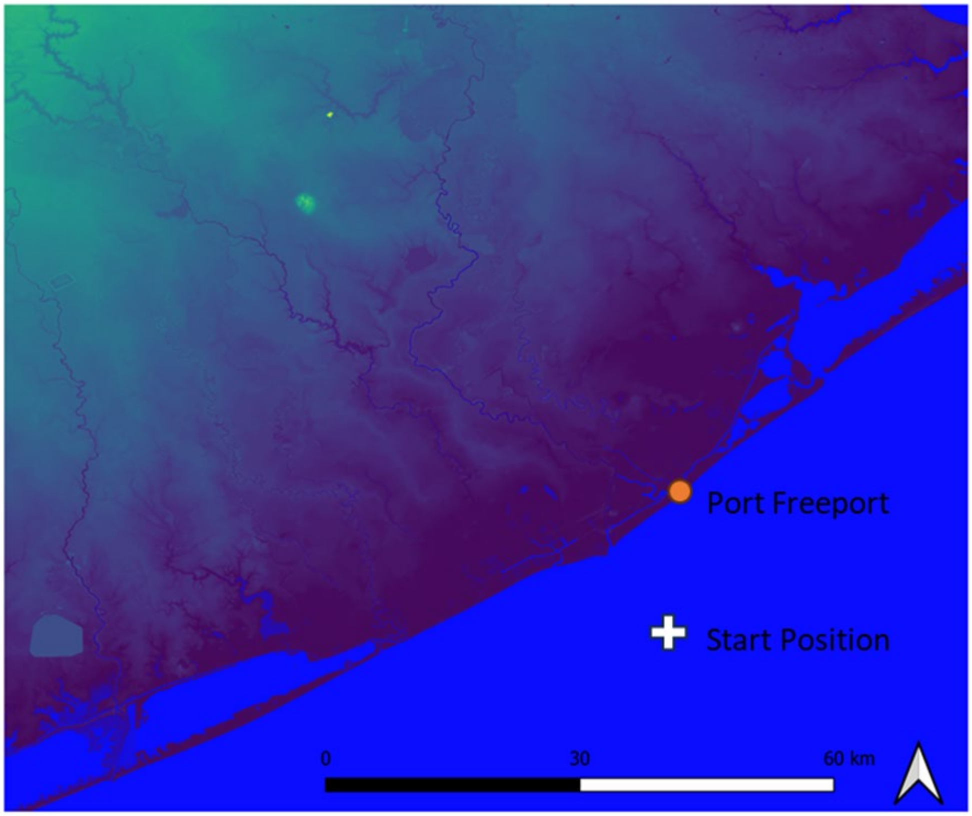

Figure 6. DEM of the landscape surrounding Port Freeport.
The orange dot marks Port Freeport, and the white cross shows a hypothetical start position of the flood-fill algorithm.

The BFS algorithm works as follows:

1. Two new storage matrices are created that have the same number of rows and columns as the input DEM. The first storage structure will be used to mark which cells in the DEM have been visited (evaluated) by subsequent steps of the algorithm (herein termed the *visited matrix*). The second storage structure will be used to store the output of the algorithm (herein the *results matrix)*, i.e., the extent of the flood and the water level above sea level.
2. A third storage structure (herein named the *stack*) is created. The *stack* stores a list of cell positions (coordinates). Cells in the DEM are referenced by a unique coordinate – row number and column number relative to the top left of the DEM.
3. A start cell is chosen that lies within the sea surrounding the port (e.g., the cross in Figure 6). In this case, the start cell is at row 12,559, column 8,420. This cell reference is added to the *stack*.
4. A loop is created that runs for as long as the stack contains entries.
5. At each iteration of the loop, the first pair of coordinates are taken from the stack, then deleted from the *stack,* and marked as visited in the *visited matrix.* Herein, this is called the *focal cell.* Note that the first iteration of the loop will take the start position identified in step 3.

6. All nine neighbors of the *focal cell* (i.e., every cell that surrounds the focal cell including diagonals) are evaluated based on two conditions:
    a. Whether the *neighbor cell* has already been visited by the algorithm (which it determines from the visited matrix**)**. If this condition is true, the current neighbor cell is ignored and the next neighbor cell is evaluated.
    b. If the elevation of each *neighbor cell* is less than the elevation of the simulated storm surge, the coordinate of the neighbor cell is added to the *stack* and is marked as visited in the *visited matrix*. This elevation is determined using the information in the DEM.

The output of the algorithm is a new matrix that represents the extent and elevation above sea level of the simulated flood (the elevation of the storm surge). Figure 7 shows two examples of this output. The depth of standing water can also be calculated by subtracting the DEM model from the output flood matrix.

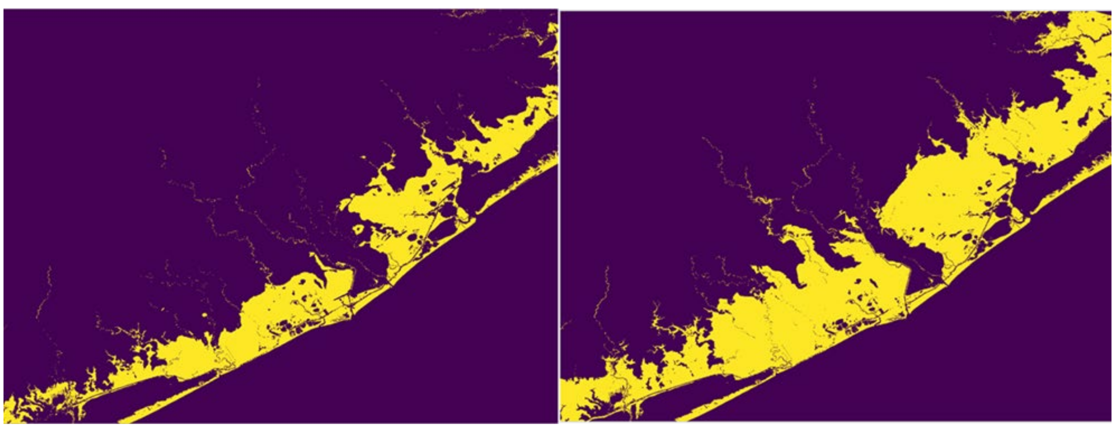

Figure 7. Examples of flood matrices created by the flood-fill algorithm.
The image on the left shows a flood extent caused by a storm surge of 2.5 meters. The image on the right shows a flood extent of a storm surge of 5m. Both images correspond to the same raster dimensions and extent as shown in Figure 6*. The flood extent is shown by the yellow cells and does not include sea areas.*

Flood Algorithm—Challenges and Extensions

The flood fill or BFS algorithm described above provides a simple way of predicting flood extent based on topology and hydraulic connectivity. Other storm surge predictors are available. For example, NOAA's Sea, Lake, and Overland Surges from Hurricanes (SLOSH) model[10] is a stand-alone software that can be used to predict flooding from storm surge and uses algorithms and data similar to the method presented above. NOAA's HURREVAC is another model that runs through a web browser, although it is available only to government emergency managers.

The motivation for our approach was to develop a tool that can be seamlessly integrated with other algorithms to measure the impact of flooding on transportation systems and to encourage further

10

development of flood models—especially those that use open-source data and algorithms. With this in mind, we suggest the following extensions to the approach:

1. The basic flood fill algorithm can be sped up by masking the extent of the ocean (or areas considered permanent water) so that these contiguous cells do not need to be evaluated dynamically each time the algorithm runs—a modification we have implemented in our algorithms. There are likely to be other ways of improving the efficiency of the algorithm for this particular application.
2. Port Freeport is protected by a levee system (Figure 8). We extended our algorithms to simulate situations where the levee remains intact and protects its internal area from flooding, or the levee's fail such that their internal area floods as per the previously described methods. To achieve this, we created a mask of the levee area and set the appropriate cells of the "visited" matrix to be visited so that any flooding would not spread to that area.
3. The flood fill algorithm can be extended to simulate both inundation of the DEM, and then subsequent drainage and dry down (evapotranspiration) to simulate the spatial and temporal change in flood extent and depth following a storm surge. During this project, we implemented this process as a separate algorithm, but for reasons of user interface complexity, we have left it out of the web-based tool shown earlier. Our implementation of a dry-down model uses a Penman-Monteith evapotranspiration model, local temperature data, and a simple water budget model (see for example the Food and Agriculture Organization of the United Nations FAO-56[11] model) to simulate temporal changes in water level as a result of evapotranspiration. The modified temporal flood method uses an algorithm to simulate drainage from flooded areas using the same concept of hydraulic connectivity used in the flood-fill algorithm, but in reverse (i.e., that allows water to flow to lower elevations, but also accounts for ponding).
4. In reality, hurricanes and tropical storms have a center (eye) surrounded by a well-defined radius of storm activity that circulates counterclockwise around the eye in the northern hemisphere. This results in variability in the height of storm surge at different distances from the eye, and because of the rotation of the storm may result in differences in storm surge height along the coast. While the algorithm defined above represents storm surge at equal areas within the region of interest, it may be possible to modify the algorithm to include vectors that simulate changes in storm surge height along the coast (Figure 9).
5. Storm surge from tropical storms and hurricanes interacts with normal tide cycles—i.e., the greatest impact on flooding occurs when storms coincide with high tides. Tidal information for different coastal locations, and information on hurricane tracks, including dates and times, sizes, and wind speeds can be found at the National Oceanic and Atmospheric Administration (NOAA)[12]. Figure 10 illustrates an example of the information available from these sites that we integrated into a series of animations during this project.
6. While it is intuitive that storm surge flooding is driven in large part by the topology of the landscape, other factors also play a role in determining flood extent and depth during tropical storms and hurricanes. For example, some storms result in high-intensity rainfall that can cause flooding via ponding. Similarly, hurricane-induced flooding can be caused by the overtopping of river channels. Storm surge can contribute to overtopping by elevating the water level along lower

[11] https://www.fao.org/3/x0490e/x0490e00.htm
[12] https://coast.noaa.gov/hurricanes/#map=4/32/-80 for hurricane tracks.

reaches of channels, thereby preventing water upstream from draining through the channel. However, overtopping can also occur when rainfall from inland areas drains into a river catchment and becomes highly concentrated in higher-order channels (i.e., those found closer to a watershed outlet).

7. Another way to extend the algorithms presented here would be to simulate different ways in which flooding can damage infrastructure. In the algorithm presented above, the focus is on modeling flood depth and temporary road closures (see next section). However, flood waters can cause permanent damage to structures such as roads. Acute damage can occur when flooding leads to excessive pressure around objects, while more chronic damage can occur when vehicles use roads and other infrastructure that have been temporarily weakened by flooding[13] (e.g., before the subsurface of roads has dried completely). As mentioned previously, we have extended our algorithms to simulate a drying phase following floods, and this would be useful for assessing the potential for chronic damage. For acute damage, more complex fluid dynamic-based algorithms could be developed to simulate water pressure and velocity outputs as well as depth.

Figure 8. Levees surrounding Port Freeport, Texas.

[13] Reference FHWA Pilot Study.

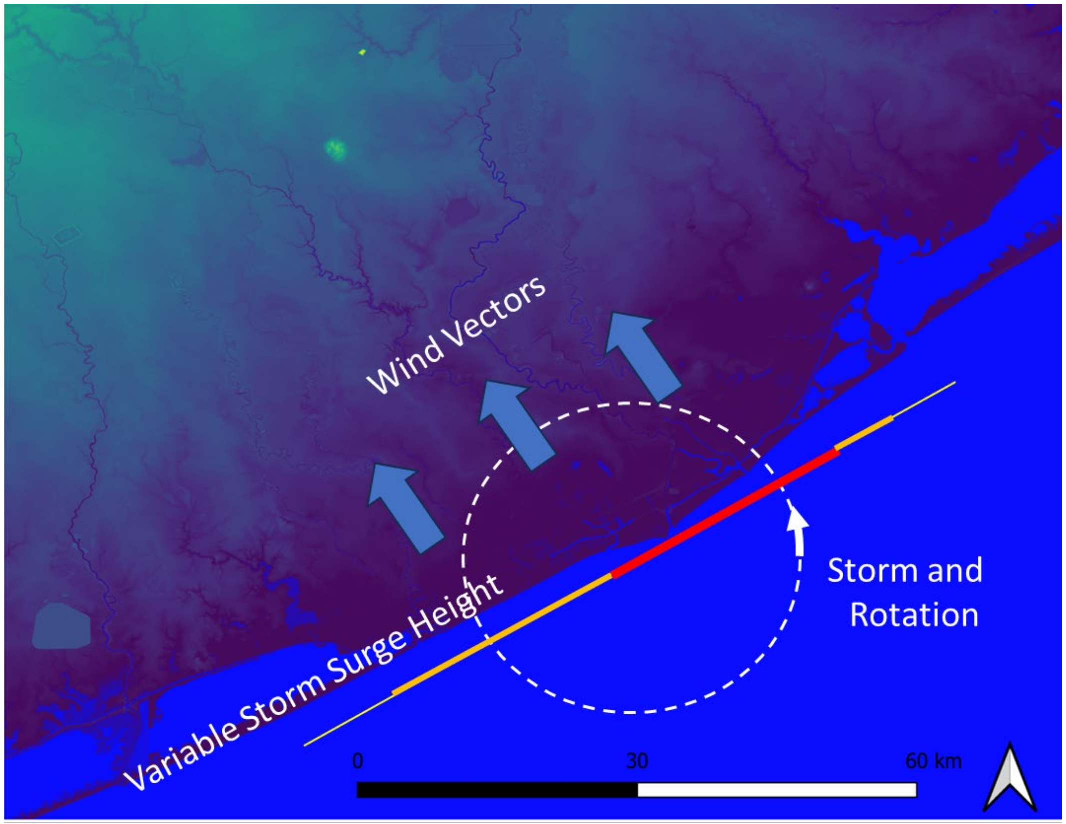

Figure 9. Conceptual model of improvements to the flood fill algorithm.
The model simulates storm surge at variable heights along the coast. The flood fill algorithm could be modified to maintain these storm surge heights in the direction of wind vectors extending inland.

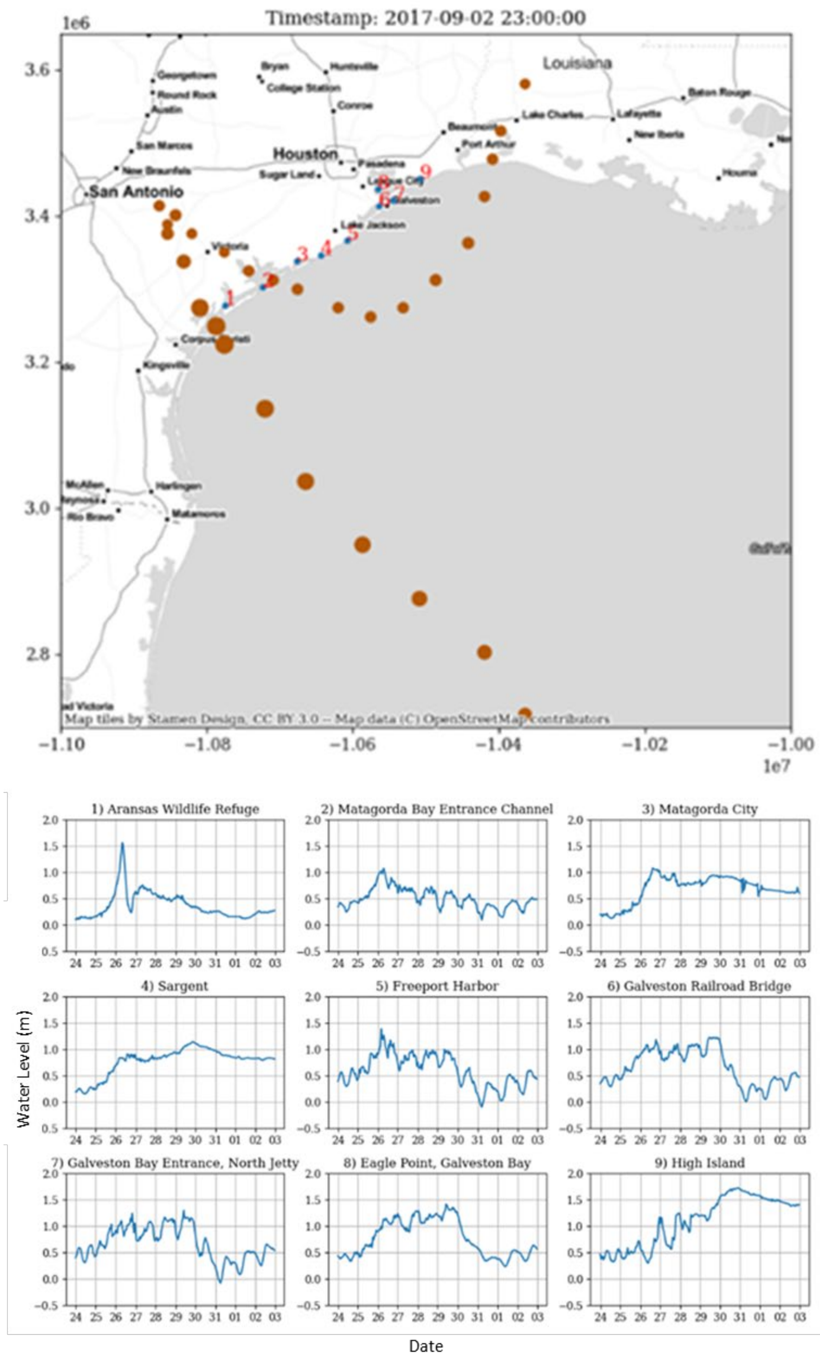

Figure 10. Trajectory of Hurricane Harvey and local tide levels at different stations along the Texas Gulf Coast.

The dotted track in the map shows the location and intensity of the storm. The nine figures show the tidal elevations at various stations along the coast.

Estimating Road Closures

This section describes methods used to calculate road closures due to flooding. The basic process for calculating road closures is to determine the elevation of roads in the area, and then compare this elevation to the elevation of flood waters defined in the previous section (based on the flood fill algorithm). Our methods used two datasets:

- OpenStreetMap road network data to provide accurate road geometry.
- LIDAR (Light Detection and Ranging) data to accurately determine the elevation of road surfaces.

OpenStreetMap Road Network Data

OpenStreetMap (OSM) is a crowd-sourced data collection and distribution initiative that maps roads and other structures worldwide. OpenStreetMap data is regularly updated by collaborators and the location and geometry of the mapped roads tend to be accurate relative to their actual locations on the earth's surface. OpenStreetMap data is also highly suitable for routing applications (discussed in the next section of this report).

We downloaded OpenStreetMap roads for Texas from Geofabrik, a website that collates subsets of OpenStreetMap data[14]. Geofabrik enables users to download data in a common GIS format as well as OpenStreetMap's proprietary format (*.osm format). For this study, we downloaded the data in the proprietary OpenStreetMap format and used custom Python code to convert the OpenStreetMap data into standard GIS formats.

LIDAR Data

We used LIDAR data from fixed-wing aircraft to estimate road elevation. LIDAR uses high-frequency pulses of light to measure distances. A common method of collecting this data is by attaching LIDAR equipment to aircraft and flying a regular flight path over a specified area[15]. During flight, the equipment sends high-frequency laser pulses toward the earth's surface. Simultaneously, detectors on the aircraft measure the reflections (returns in LIDAR terminology) of these pulses from the earth's surface and match this with accurate GPS coordinates of the aircraft's location. The resulting data is processed to provide the spatial location of each return as well as information such as the wavelength of the reflected light, and the time interval between laser pulses and each return. Finally, these measurements are used to infer the elevation, and to some extent the material properties of objects that the laser pulses were reflected from. Aerial LIDAR is capable of providing high-resolution information on the height of all objects on the earth's surface that are over a size threshold determined by the resolution of the LIDAR sampling. This includes vegetation, buildings, utility equipment, and roads.

Aerial LIDAR data is often collected in flood-prone or coastal areas because the data provides high-resolution data on the elevation of objects which is useful for flood prediction and prevention. LIDAR data

[14] https://download.geofabrik.de/north-america/us.html

[15] Other methods include LIDAR equipment mounted to vehicles to obtain fine scale information about road surfaces. The United States National Aeronautics and Space Administration (NASA) also has a program (GEDI) that collects LIDAR data from space.

is often collected by public agencies, and as such, is often publicly available and free to use. We downloaded LIDAR data for the Port Freeport area via the USGS's National Map webpage. As Figure 11 shows, the USGS houses LIDAR available for large areas of the US. Other sites that distribute LIDAR data include OpenTopography (https://opentopography.org/)and the U.S. Interagency Elevation Inventory (a collaboration among USGS, NOAA, the Federal Emergency Management Agency, U.S. Army Corps of Engineers, and others). Many metropolitan planning organizations also curate their own LIDAR data.

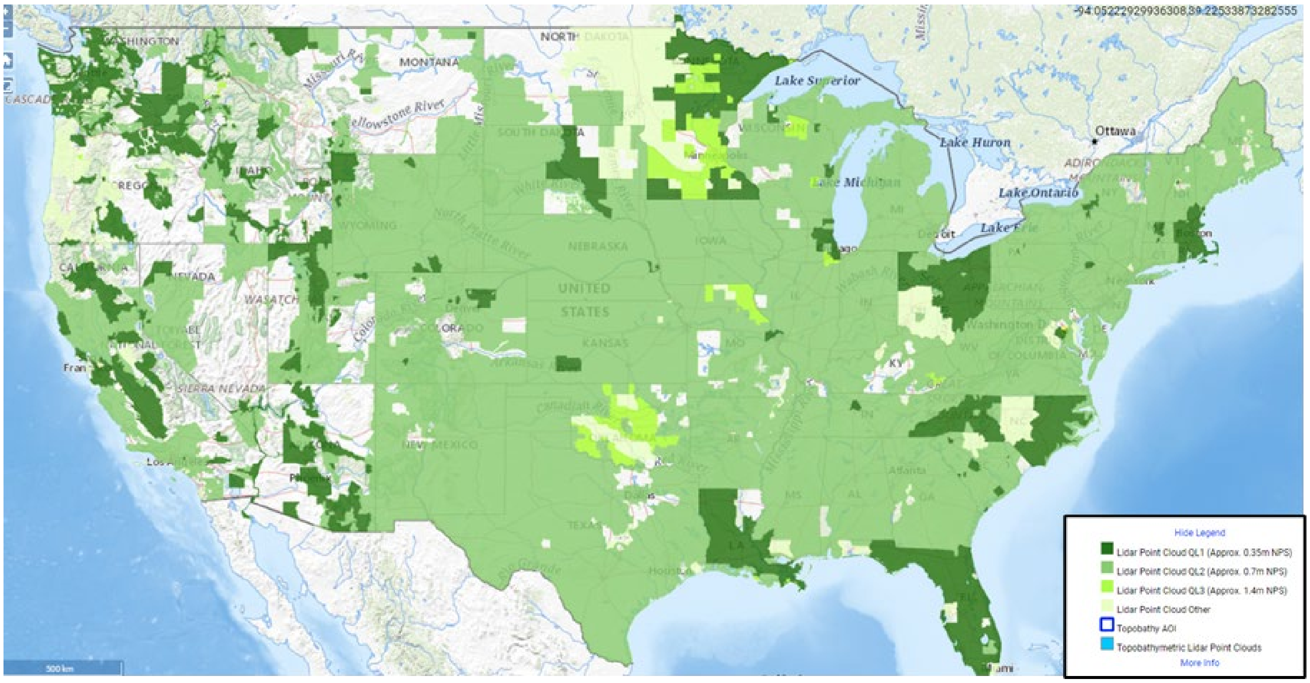

Figure 11. Map showing the availability of LIDAR data for the United States as provided by the USGS National Map[16].

LIDAR data is most often provided in the form of a GIS raster describing elevation or point cloud data. LIDAR-based raster data have already been processed to provide a representative elevation in each cell of a raster (similar to the DEM models described in the previous section). As such, the precise methods behind the processing determine exactly what type of surface elevations are described by the resulting elevation raster (for example, a bare earth DEM, or layers that have been analyzed to specifically represent buildings and human-made structures, or vegetation layers). LIDAR data made available in the form of point clouds may also have been preprocessed, but this data format remains closer to the nature of raw data. Figure 12 shows an example of LIDAR point cloud data downloaded for the Port Freeport study area. Each point shown in the figure corresponds to a LIDAR return as described previously. The subset of LIDAR data shown in the figure measures 1.5 km x 1.5 km and consists of 23,966,016 points or LIDAR returns (equivalent to 10-11 points per m^2).

[16] *Taken from https://apps.nationalmap.gov/lidar-explorer/#/*

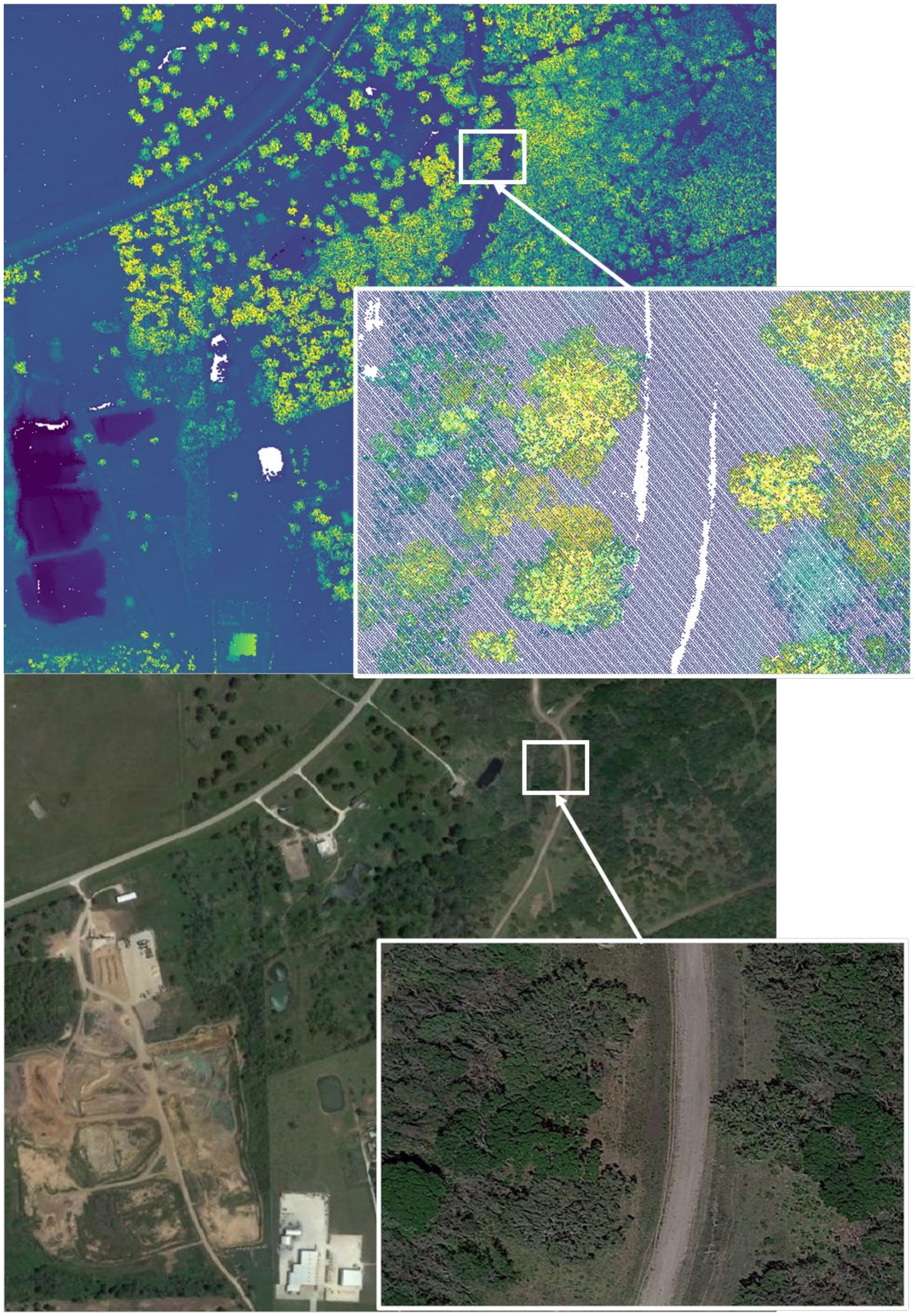

Figure 12. Example of LIDAR point cloud data compared to aerial imagery data.

The top main map shows LIDAR point cloud data for an approximately 1.5 km x 1.5 km area within the Port Freeport study area. The inset map shows point cloud detail for an approximately 100 m x 100 m area. Each image comprises a large number of points colored by elevation. The lower images show aerial imagery of the same locations.

Calculating the Elevation of Roads

The elevation of roads in our study area was calculated using the aforementioned LIDAR and OpenStreetMap data. This was done by opening a GIS version of the roadway network and then iterating through each link in the file and performing the following steps on each link:

1. For each link, iterate through each vertex of the road geometry. Each roadway link is defined by two or more vertices (i.e., points where two or more lines meet). At the minimum, each link contains two vertices defining the beginning and end of a link, but in most GIS formats, curves and other roadway shapes are represented by a larger number of vertices that approximate the actual geometry of roads.
2. Draw a 12-ft buffer around each vertex and calculate the 5th percentile elevation of all LIDAR points that fell within the buffer.
3. Save the location of each vertex to a point-based GIS file and label the vertex with a unique road link ID.

Determining Road Closures

Combining LIDAR-based elevation with the OpenStreetMap road data resulted in a GIS file (of point geometries) containing the estimated elevation of each vertex of every road link in our network as well as a unique road link ID that specified which of the original OpenStreetMap links the vertex belonged to.

The final step in the process of determining road closures uses the flood layer described in the previous section, the original OpenStreetMap road link data, and the derived road vertex elevation data. The following processing steps were undertaken:

1. The flood layer derived from the flooding algorithm was converted from a matrix (raster) representation to a vector (polygon representation).
2. The resulting flood polygons were used to query all road vertices that fall within the boundary of the simulated flood and that also have an elevation less than the elevation of the flood.
3. The subset of vertices is used to determine a list of the original OpenStreetMap road link IDs that are affected by the simulated flood (note that this assumes that if any vertex on a link was flooded, then the whole road is affected.).

Each of the processing steps above was undertaken using Python. The GeoPandas library[17] was used for GIS processing. This library supports spatial indexing and spatial queries which made the selection of flooded vertices relatively efficient. The efficiency of spatial indices was the principal reason the original raster flood layer was converted to a vector (polygon representation). The researchers used the Python OpenCV library[18] to perform this conversion.

Road Closure Models – Challenges and Extensions

Before considering extensions to the road closure models it is important to stress that both the extent of flooding (as discussed in the previous section) and the likelihood of road closures are complex processes.

---

[17] https://geopandas.org/en/stable/
[18] https://pypi.org/project/opencv-python/

In reality, roads are hydraulically engineered to minimize flooding and to provide hydraulic connectivity within the broader landscape. As such, predicting flooding and road closures is a complex task that involves reciprocal interactions between the built and natural environment.

Predicting road closures also involves a human decision-making dimension. For example, how much standing water does it take to close a road? And that human dimension may change depending on context. While the general public is becoming more educated about the dangers of any standing water on a road surface, some flooded roads may be passable to emergency or specialized vehicles if travel needs are great enough.

Given the scope of this study, we have developed simple models, based on intuitive ideas. As with all models, their utility depends on knowing how and when to use them and interpret their results. With this in mind, the researchers suggest several ideas that could be used to extend the current road closure model and data:

1. Aerial LIDAR data has many potential uses and is becoming increasingly common. In this study, we have used it to estimate road elevation because road elevations are not accurately represented in DEMs. However, in the context of flooding and road closures, LIDAR could also be used to detect the location of trees or vegetation that could be windblown during hurricanes. LIDAR could also be used to detect important utility structures such as electricity lines and poles. Since LIDAR data is often repeatedly collected for the same areas, it could simply be used with aerial photography to visualize changes in the topography and development around a port.
2. Perhaps the biggest challenge (and therefore most worthwhile extension) of flood and road closure models is to involve hydraulic specialists from local flood control districts. The potential barriers to this are time, funding, and mechanisms through which effective collaborations can be undertaken. As mentioned previously, effective resilience thinking will require the boundaries of systems to be redefined, or for the behavior of existing systems to be understood more clearly. Models and interactive data visualizations are an effective way to initiate collaboration between experts because they promote effective discussion on what processes should be included in a model (i.e., elements of simplification), judgments on the predictive ability of models relative to expert knowledge, and suggestions for how a model can or should be used given its predictive ability.

Routing Models

In this section, we describe the models and methods used to estimate the impact of flooding on port-hinterland transportation. The approach we use to assess this impact is to calculate (quantify) changes in travel cost (e.g., travel time, travel distance) that may arise as a result of road closures.

We used graph theoretic[19] data structures and routing algorithms to calculate travel costs. Routing algorithms are widely applied in transportation science and are found in applications such as Google Maps (or other web mapping applications), freight logistics software, and travel demand models. These algorithms and models extend more formal fields of mathematics and computer science called graph theory. Thanks to initiatives such as OpenStreetMap and several programming libraries that provide standard algorithms, routing algorithms are becoming increasingly available to research scientists with limited expertise in the mathematics and science of graph theory.

The first step in the development of our routing process was to download OpenStreetMap data for our area of interest. We downloaded data for the whole of Texas and then clipped it to a region extending 100 miles (~160 km) from Freeport (Figure 13). OpenStreetMap data contains information on all roads and paths, from interstate to pedestrian trails in a format ready for routing algorithms. We limited our representation of the road network to roads that vehicles can travel on – including residential roads, but not including service roads (in OpenStreetMap terminology these are roads that makeup parking lots, industrial units, or other such entities). One of the limitations of OpenStreetMap data is that roads do not always contain information on speed limits, the number of lanes, or other information useful for this study. To overcome this, we matched OpenStreetMap roadway links with the Texas Department of Transportation's GRID database of Texas roads to obtain speed limit, number of lanes, and other information for each road.

After cleaning and processing the OpenStreetMap data, we used the iGraph Python library to convert the individual road links into a bi-directional graph ready for the application of routing algorithms. Figure 14 illustrates a bidirectional graph representation of a small, hypothetical road network. The nodes marked by letters represent road junctions. The arrows (directional edges in graph terminology) represent possible travel directions, and the numbers next to each line indicate the cost of travel for each direction (e.g., travel time). For example, in Figure 14, the shortest path from node A to I is through nodes B, C, G, and F. If the numbers next to the links represent travel time (in minutes), this path would result in a travel time of 31 minutes. Note however that the nodes and edges between B, C, G, and F signify that travel is possible in one direction only such that the shortest path from node I to A is through nodes F and B, yielding a travel time of (40 minutes). Note also that an additional path exists between nodes A and I, through nodes B, D, and H. When road networks are represented in this way, established, proven algorithms can be employed to efficiently find the shortest path through networks containing many millions of road sections (links) and junctions.

[19] We have chosen this terminology to distinguish between the more conventional meaning of a graph – i.e., a scientific figure.

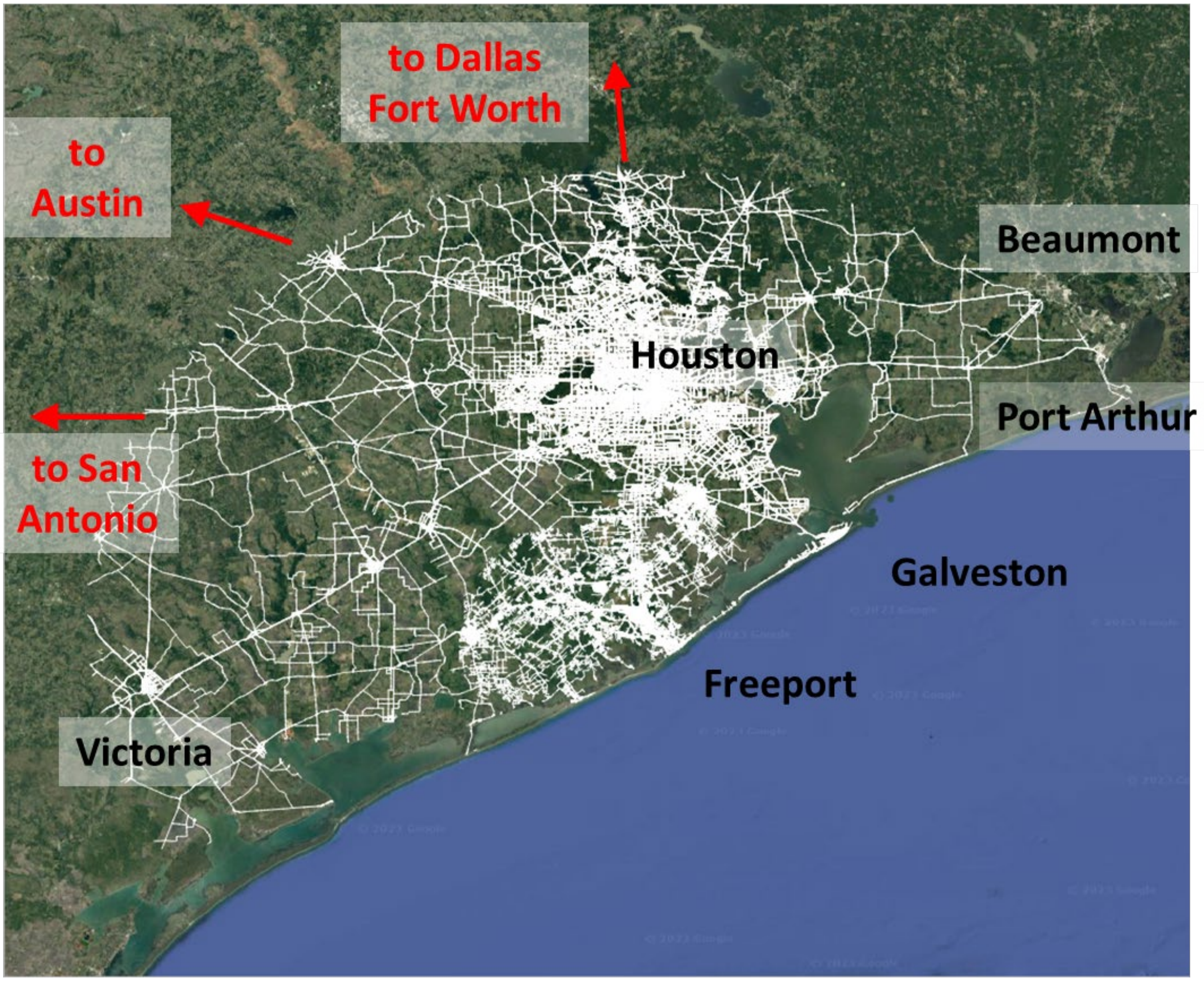

Figure 13. OpenStreetMap data that were used in this study.

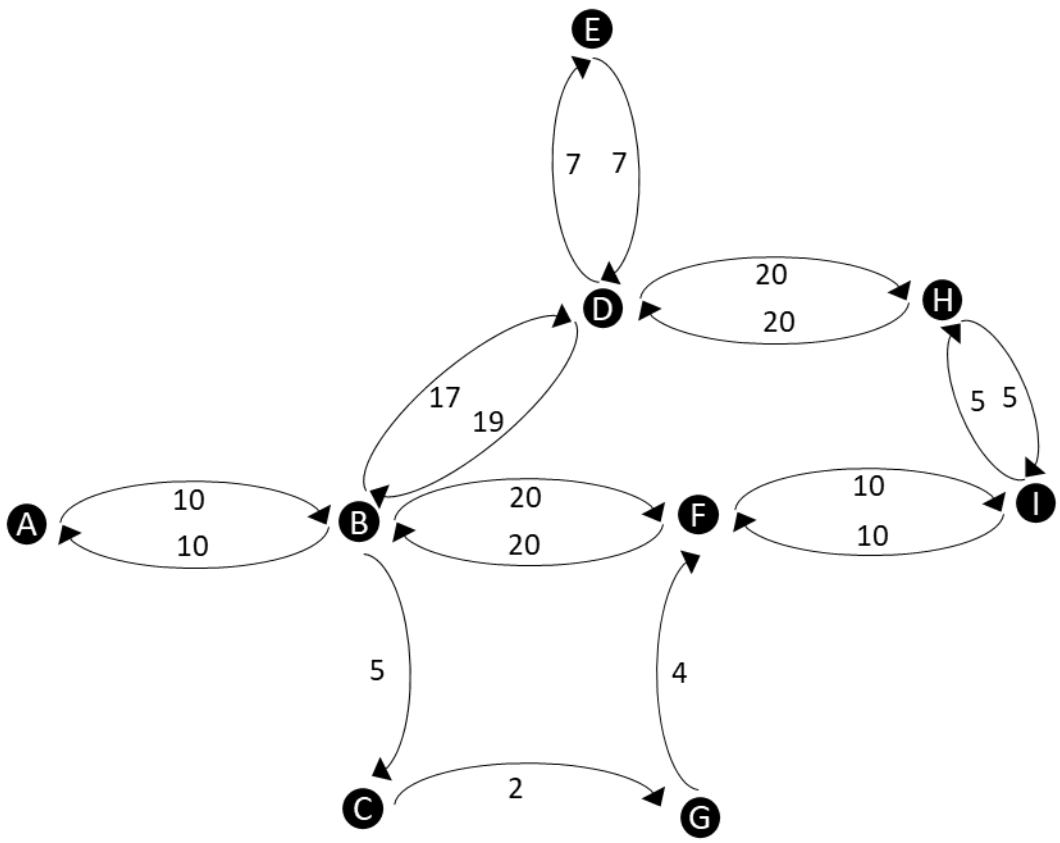

Figure 14. Example of a graph theoretic representation of a bidirectional road network.
The nodes labeled letters A through I are connected by edges that specify possible routes through the graph. The edges are directional, such that if only one edge exists, travel can only proceed in that direction. The numbers beside the edges represent the "travel cost" that can be used to find the shortest or most efficient path from a specified start node to a specified end node.

Implementation

The creation of a graph theoretic representation of the network is straightforward using iGraph or alternative graph libraries. First, travel times and other travel cost metrics were calculated for each road link using the length of the road section and an estimate of speed (in this study we used posted speed and link length to calculate travel time). Each node (junction) and edge (road section) were then given a unique ID (the same ID as those used in the previous section describing road closures). Finally, functions within the iGraph library are called to initialize the graph, populate it with each node, and specify which edges connect each node. The road network used in this study (shown in Figure 14) consists of over 318,000 road sections and 600,000 nodes. Using our approaches, equipment, and the Python iGraph library, the road network takes only 1.4 seconds to initialize as a graph.

The purpose of the routing models is to calculate a change in travel cost for a disturbed and undisturbed network. To accomplish this, two graph models were created—first an undisturbed network, i.e., the road network based on the original OpenStreetMap data, and second, a disturbed network created by removing (or increasing the travel time) for the flooded roadways identified by the methods described in the previous sections.

The final ingredient in the routing process is to find the shortest paths through the undisturbed and disturbed networks using inbuilt and custom routing algorithms. The web-based application described previously provides three different implementations of routing algorithms.

1. ***Custom Routing:*** The custom routing method uses a single start location and calculates the shortest path between them using Dijkstra's shortest path algorithm. In all cases, a representative node close to Port Freeport is hardcoded as a start location. In the web application, a custom destination location is specified by clicking the web-based map. This initiates a search for a node nearest to the clicked point (i.e., based on the GIS coordinate system) that serves as the destination node. Both nodes are provided as inputs to iGraph's built-in Dijkstra's shortest path algorithm. This algorithm returns the shortest path through each graph (essentially an ordered list of nodes that represent the shortest path). This information is then used to find the total travel cost of the shortest path.
2. **Regional Routing Impacts:** This algorithm finds the shortest paths from a single source origin or destination (Port Freeport) to all other nodes in the network. The travel times for each node and link are then compared and the results are presented as a colored map showing areas relatively unaffected by the specified disturbance and those relatively affected by the disturbance Figure 15).
3. **Routing based on Min-Cut Algorithm:** All shortest path algorithms require a start and a destination location (or nodes) to be specified. For many ports, key routes, such as those that link the port to major warehouses or facilities will be known and can be explored through the custom routing tool. But what if the key routes to a port are unknown, or if a general method is required to assess all routes to a port? The min-cut algorithm attempts to address this problem by automatically finding all key links to a port. Specifically, the algorithm finds the minimum set of links that vehicles must travel through to reach a specified 'inner zone' surrounding the port assuming that all vehicles begin their journey from outside a specified outer zone (Figure 15). The details of the algorithm along with an example of its application to port security are discussed in Appendix 1. Once these key links are found, conventional shortest-path algorithms are run to find travel time differences for all key locations.

A common approach for each routing algorithm is to use a single destination location (i.e., Freeport in the case study) and to vary only source locations (or vice versa the port can be thought of as the origin). This greatly simplifies the routing methods and the user interfaces necessary to customize routing.

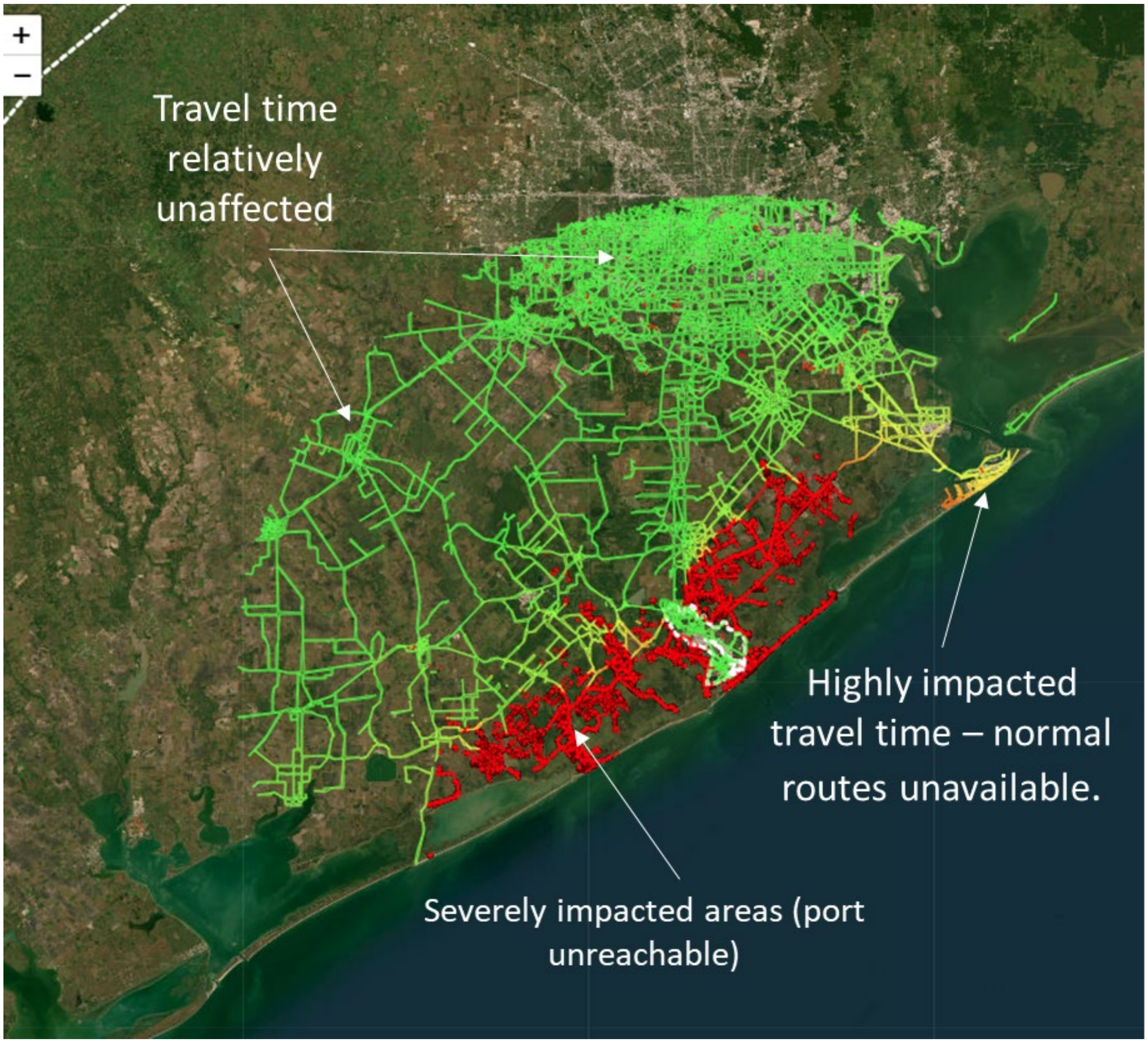

Figure 15. Example of regional travel impact analysis for flooding.

Flooding was predicted from a 5m storm surge. The red areas show areas where traffic to and from the port is not possible. The yellow areas show areas where travel time to and from the port is severely affected by road closures, and the green areas show areas unaffected by road closures. Note how, in the case of Port Freeport, the levees around Freeport (white line) maintain routes into the port and therefore keep most of the hinterland accessible.

Figure 16 illustrates the result of using the Min-Cut algorithm to find key road links for vehicles traveling to/from Freeport. The key road links (numbered 1 to 12) represent a minimum set of links that vehicles traveling from outside the yellow area must pass through when traveling to any location in the inner zone (the area inside the yellow shading).

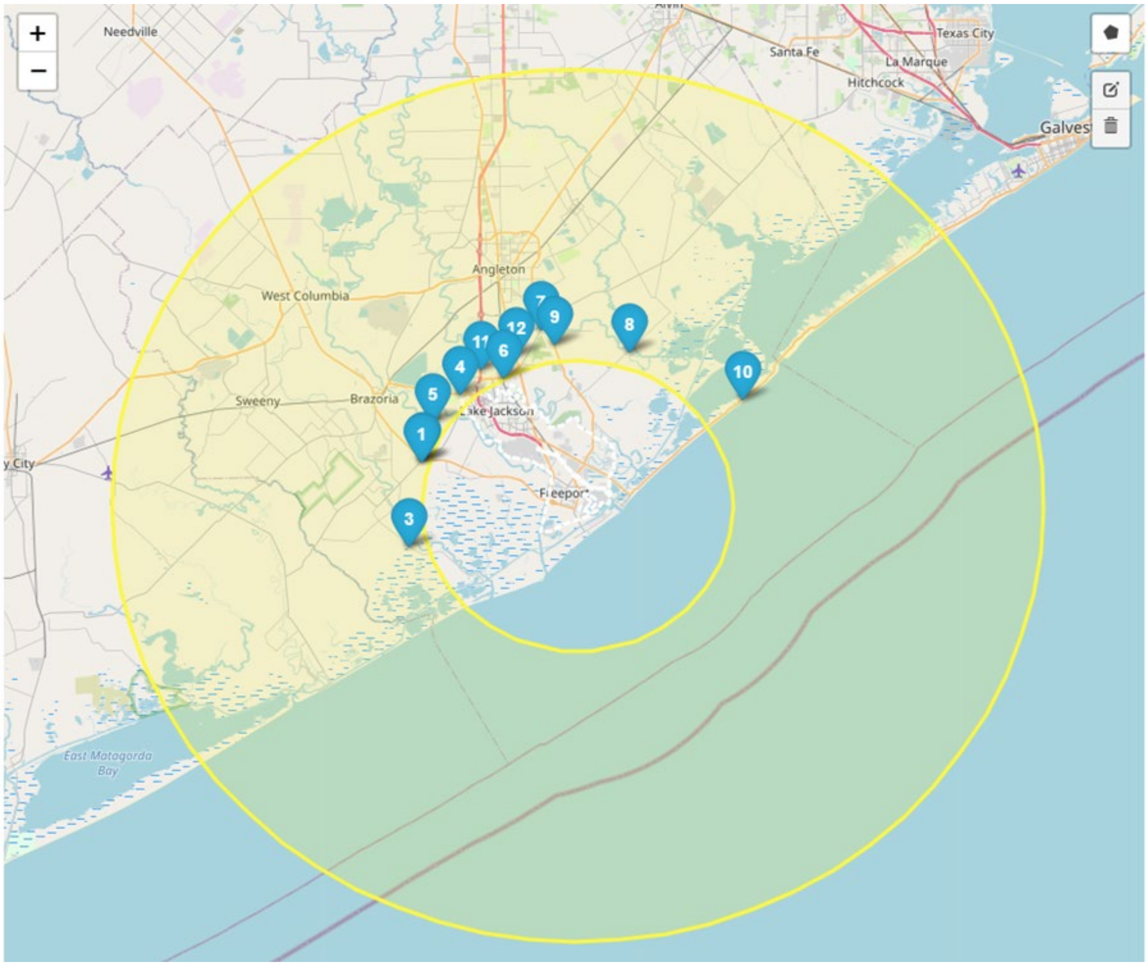

Figure 16. Result of Min-Cut algorithm to find key road links for vehicles traveling to/from Freeport.

Routing Models - Challenges and Extensions

The availability of OpenStreetMap data and a wide range of graph software libraries available for a range of programming languages make it easy to implement simple routing algorithms. The largest technical challenges that arise from using these data and algorithms lie in the need to preprocess and clean data for a specific application, and in understanding and using Graph libraries to efficiently perform routing. For example, we preprocessed the OpenStreetMap data by matching those links with the geometry of links described in the Texas Department of Transportation's roadway data set (GRID). This is a challenge because the geometry of roadway sections in the two data sets do not always match. The second technical issue arises because graph libraries implement shortest-path algorithms in different ways. To achieve fast execution times, it is often necessary to experiment with the built-in algorithms of a library to find the fastest way to perform a routing analysis for a specific application. The abundance of theory and documentation for graph algorithms, as well as the availability of open-source libraries, greatly helps these efforts.

Instead of using custom algorithms, it would be possible to implement third-party routing algorithms without much programming expertise. For example, Google Maps implements sophisticated routing algorithms through point-and-click web interfaces. Google also provides an API (Application Programming

Interface) that enables these same algorithms to be run repeatedly with only minor programming. GraphHopper[20] offers a routing API but with the added benefit of maintaining open-source code that provides options for customization.

As with previously discussed algorithms, in this study, we have intentionally taken a hands-on approach to developing routing algorithms. This provides opportunities for extensions to a particular application and helps build a fundamental understanding of approaches (of benefit even if one were to incorporate existing off-the-shelf APIs).

The following list details challenges we faced in implementing our approach, or ideas that could be used to extend or improve our methods and algorithms:

1. We implemented two methods to calculate the difference in transportation cost between a disturbed and undisturbed route. The first and simplest method uses the difference in travel time between the whole trip (defined by origin and destination for the two routes). This is then used to calculate the loss of service as a percentage of the total travel time for the route. One problem with this approach is that a detour of a given length will represent a large percentage of a relatively short trip, but only a small percentage of a large trip. To overcome this effect, we created a version of the routing algorithm that calculates the increase in travel time for only the ***detoured*** portion of a trip. To do this, we calculated the portion of the disturbed route that deviated from the undisturbed route, thereby finding the first and last link of any detour from the undisturbed route. Then we calculated the travel time between these two links for the undisturbed route vs. the disturbed route and expressed the difference as a percentage. This small exploratory extension to the model highlighted the need to think carefully about how to measure travel impacts.
2. In our application, we used a bidirectional graph to represent the network. Real-world road networks are inherently one-directional—for example, they often contain important one-way routes, and travel times are not always equal in both directions. However, bi-directional graphs add considerable complexity to the task of developing algorithms, as well as reducing their speed. For an application such as the one presented in this project (which is essentially a sketch planning application), it may be fruitful to explore how to simplify a network into a simpler undirected graph structure (i.e., where each edge represents travel in both directions simultaneously).
3. In the real world, the routes that drivers take through a network have distinct behavioral components. Regular passenger car drivers may choose routes that they prefer based on a criterion other than travel time – such as aesthetics, ease of travel, reliability, or opportunities to perform other tasks within a trip or journey. Heavy-duty vehicle drivers might stick to certain routes until the first and last miles and must take precautions to avoid vulnerable structures such as bridges. Exploring methods to add this realism into basic routing algorithms is likely to be a profitable exercise – especially if there is a need for custom routing algorithms for specialized, niche research projects.
4. Our routing algorithms are applied only to the road network. It would be relatively easy to extend these algorithms to include other modes, such as rail. For port applications, the addition of water

[20] https://www.graphhopper.com/.

routes may also be fruitful. For example, an analyst could model or explore options for maintaining essential access to a port if all terrestrial links become disabled[21].

5. Opportunities exist for general research into the application of graph theory towards transportation network analysis. Shortest path algorithms are well-known and well-used in the field of transportation. They also provide almost unlimited opportunities for extension and improvements. However, graph theory provides many more methods that are generally useful for analyzing roads- or other networks). The Min-Cut algorithm is an example of extending the use of formal graph theory to the transportation domain. Other methods and algorithms that may have application to niche routing applications most likely already exist. The challenge in being able to discover and apply these algorithms is to find experts in both fields willing to collaborate and experiment.

Integration and User Interface

We integrated our algorithms into a web-based GIS application (Figure 3 through Figure 5). The idea was to explore the feasibility of an application that could be used in real-time to investigate different storm scenarios, their effect on road closures, port-hinterland transportation service, and therefore resilience. We imagined the application could generate useful collaboration and discussion among stakeholders and local experts in different domains. We envisaged that the highly quantitative outcomes of the tool, and the simple but understandable models, could encourage useful challenges and discussion by local experts to generate new ways of looking at the system.

Integration and User Interface - —Challenges and Extensions

We encountered three main challenges in integrating these algorithms into a web-based application. The first was the speed or runtime of each component of the application; for use in real-time, the speed of our algorithms was too slow. Specifically, a web-based approach requires extra time to deliver the results of any algorithm over the web onto a client web page. While the run time of technical portions of the algorithms was often acceptable, the extra time taken to visualize results resulted in slow performance.

The second challenge of the web-based approach involved the complexity of the user interface. Our existing prototype application was designed to integrate the algorithms rather than details from the perspective of a user. Nevertheless, even in this simple form, the web-based approach was challenging to develop in a way that allowed full usability. For example, as mentioned previously, we struggled to integrate a dry-down model that would have added a useful time component to the analyses.

The third challenge arose from the need for simultaneous users to access the web application. Although our application is never likely to attract large numbers of simultaneous users, web applications must be programmed to deal with two users at the same time. The challenge here is that for every simultaneous user of the system, copies of data (e.g., predicted flood layer, road closures, and disturbed network) must

[21] For example, to an offsite store of emergency supplies to maintain port facilities, or to supplies for sustaining port communities in the event of major disaster.

be saved for the unique scenarios they are exploring. This adds technical programming difficulties but also increases the memory requirements of servers hosting the application[22].

Possible extensions and improvements to the methods of integrating the flood and routing algorithms include:

- Developing an integration within an existing, free, open-source desktop GIS application such as QGIS. QGIS enables developers to develop add-ins that build upon the base functionality of the software but are targeted to a niche audience or a specific task. Integration through a desktop application may overcome some of the challenges of developing a web application, without sacrificing accessibility to stakeholders interested in the approach.
- Integrate the algorithms without a user interface. The data and models presented in this project could be used in a targeted research project that is more focused on more specific flooding, routing, or resilience questions (instead of the technology development goals of this study). Such approaches could leverage some of the methods and data described in this report, for example, to conduct a comparative analysis of transportation resilience among different ports.

[22] This is one reason this kind of routing application requires a custom approach rather than relying on an existing API from an existing provider. Established APIs are designed to provide routing for the masses, using a relatively consistent network, rather than provide routing under unique and specific assumptions such as the ones explored in this project.

Conclusions

This project evaluates the availability of data and models that can be used to help predict the impact of floods on regional transportation links, and therefore the resilience of port-hinterland transportation systems to storm surge flooding.

The project used a case study approach to evaluate data and models and integrate them into an IT tool that could be useful for exploring concepts of resilience. While the case study focuses on Port Freeport, Texas, the models and data were chosen such that the tool could be expanded to other ports, and for other study approaches.

Our endeavors suggest that there exist considerable amounts of publicly available data useful for evaluating the resilience of port-hinterland transportation systems. Some of this data may be largely unknown to transportation practitioners but may offer many opportunities for research into port systems. For example, aerial LIDAR data is becoming increasingly available through national data collection programs and has the potential to be used to develop high-resolution models of the earth's surface useful for hydraulic engineering and object detection. Similarly, OpenStreetMap data represents a comprehensive map of roadways and other structures, that when paired with open-source programming libraries provides opportunities for novel routing algorithms– especially for niche applications such as this study. We hope that this research instills confidence in other researchers to explore these opportunities and provides a baseline for how to begin these investigations.

Our research suggests that it is possible to extend our data and models to develop operational applications to quantify the resilience of port operations to disturbances to its hinterland transportation network. We suggest that an important step in any extensions of the models and data would be to collaborate with hydrologists and other specialists to ensure the latest methods and models are incorporated or simplified in a way that provides meaningful flood predictions from a variety of flood drivers (e.g., storm surge, channel overtopping, ponding), and that have the potential for predicting a range of causes of damage to the network (e.g., effects on road damage as well as effects on temporary road closures).

In the context of resilience and resilience thinking, we suggest that our approach of developing and integrating open-source data and algorithms is a potentially useful way of promoting effective collaboration between hydraulic engineers, transportation specialists, port managers, and other stakeholders. As previously mentioned, data and models already exist that are useful for exploring the interactions among hurricanes, flooding, road closures, and transportation impacts. The integration of these models and data into a practical tool provides a useful construct to explore how data, models and methods can be appropriately simplified to achieve a specified design goal.

Appendix 1 – Calculating Origins and Destinations Using Minimum Cut Algorithms

This appendix details a method to calculate a minimum number of network links for vehicles traveling to a specified destination—for example, a port. Specifically, the method makes it efficient to calculate a minimum number of network links (and their location) that **all** vehicles traveling toward the port must pass through when traveling from outside a specified zone to another specified zone. Formally, the resulting links are often called a minimum cut set.

Figure 17 illustrates the concept of the minimum cut set for two different road networks defined by increased spatial scale. The left panel shows a hypothetical location (e.g., a port) on the network (orange dot). The dashed circular lines define an outer and inner zone. In the left-hand panel, all vehicles traveling from the outer zone to the inner zone may pass through the links marked as red or yellow. However, the minimum cut set of links is shown in red. These are the minimum set of links that all vehicles must pass through when moving from the outer zone to the inner zone (and vice versa in a fully bidirectional network). As shown by the left-hand panel, minimum cut sets are relatively easy to find or calculate for small networks. However, for larger networks, such as the hypothetical case shown in the right-hand panel, determining the min-cut set manually is challenging.

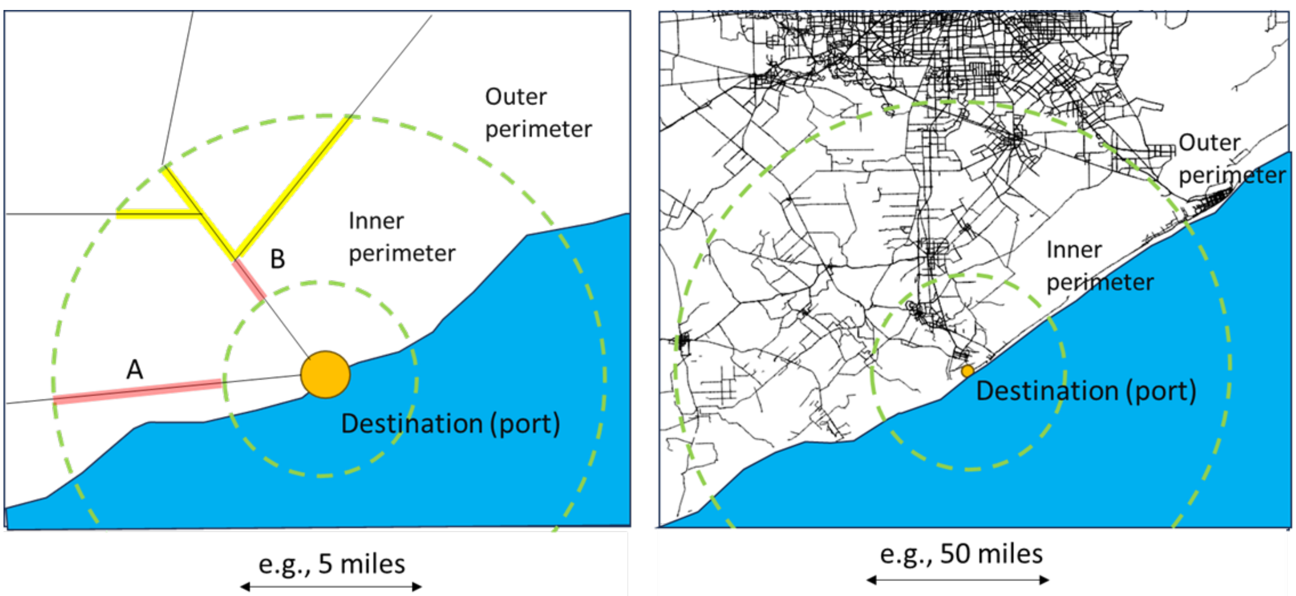

Figure 17. Examples of minimum cut sets for a road network.

Computer algorithms (minimum cut set algorithms) exist to find the minimum set of roadways that vehicles must pass through to reach a specified zone within a network. The algorithm used to calculate the minimum cut links used in the network analysis described in the main body of the report was inspired by a paper by Anderson et al (2007)[23]. The authors describe methods to investigate the feasibility of monitoring all potential routes into the city of New York to prevent terrorist attacks. Since monitoring any location involves cost, they are motivated to find the smallest possible set of locations that could be used to monitor all vehicles entering the city. Figure 18 shows the network used in their analysis. The network is complex and highly connected and contains 488,951 road sections. Despite this complexity, their methods found that all traffic that enters New York City must pass through only 89 critical locations.

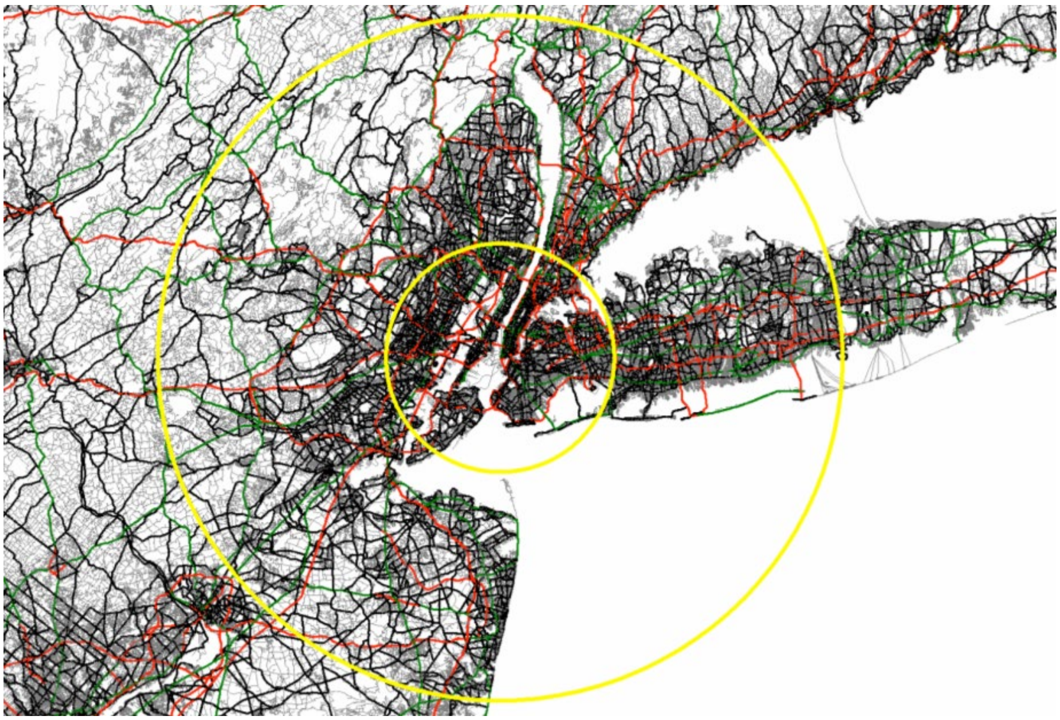

Figure 18. Road Network surrounding New York city.

The steps below outline the method used to obtain the minimum cut set (also illustrated in Figure 19):

1. Two areas of interest were defined. The first defines the maximum extent of the analysis, i.e., how much network is used in the final routing engine. This area of interest defines the maximum distance between the outage and any calculated Origin-Destination (OD) (min cut) locations. The second, smaller area of interest defines the minimum distance (from the road closure) of the required O-D locations.
2. All roads that are completely within the smaller area of interest are removed from the network. The terminal ends of each road that crosses the inner area of interest are connected to a 'super node' (red node shown in Figure 19 panel 3). All roads that cross the second area of interest are connected to a second 'super node' (the blue dot in Figure 19 panel 3). Roads between the inner and outer areas of interest are given an edge capacity of 1. All edges connecting the

[23] Barnett, R. L., Bovey, D. S., Atwell, R. J., & Anderson, L. B. (2007). Application of the maximum flow problem to sensor placement on urban road networks for homeland security. Homeland Security Affairs, 3(3).

terminal ends of the inner and outer roads to the super node are given a very large edge capacity (in the original method by Anderson et al they use a value of infinity, but that was not possible in our case). A new edge (with infinite capacity) is used to connect the inner and outer super nodes.

3. Finally, a maximum flow/minimum cut algorithm is run on the network. This algorithm returns the minimum cuts for the network and at the same time the minimum number of roads that all vehicles must travel through towards the inner area of interest. The research team used an algorithm provided by the Python library iGraph (https://igraph.readthedocs.io/en/stable/).

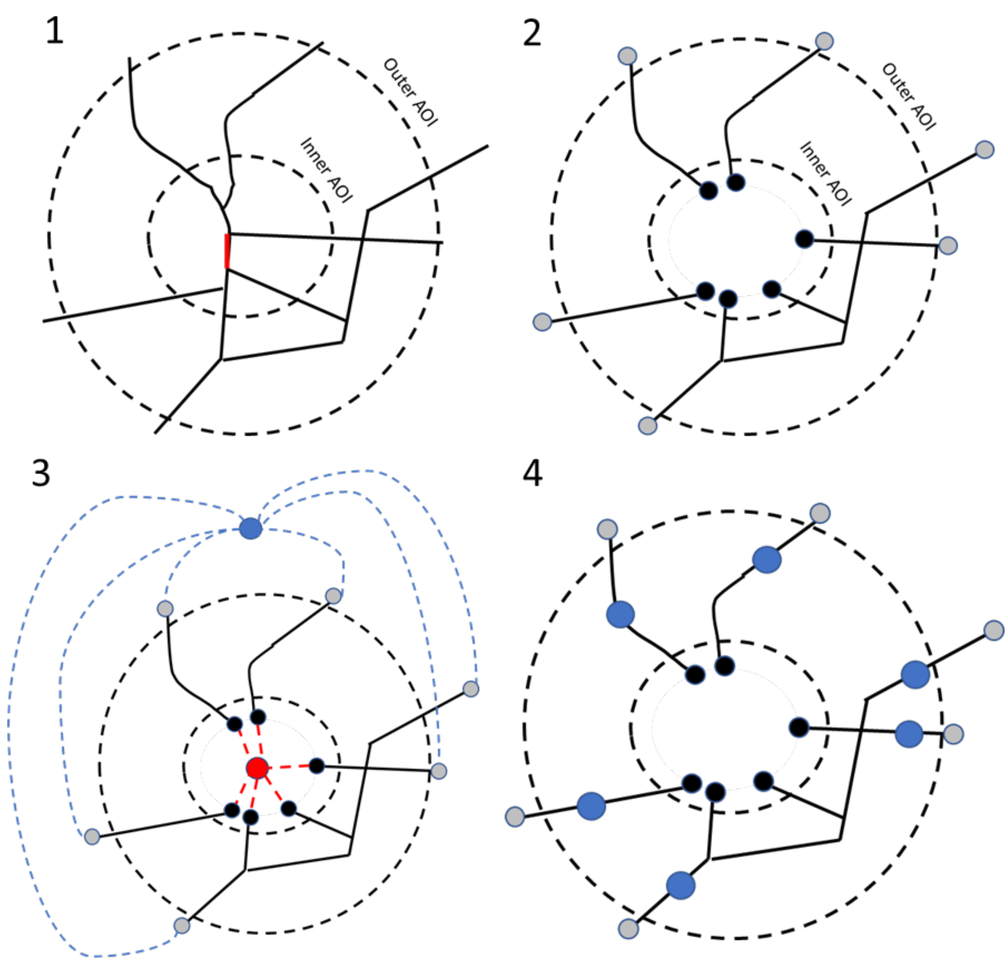

Figure 19. Illustration describing the steps to calculate minimum cut sets of origin and destination points used in the network analysis.