IEMS 393: Senior Design Project

Integration of Real Time Data in Urban Search and Rescue

Alex Huang
Alex Ma
Sara Schmidt
Nancy Xu
Brandon Zhang

March 8, 2012
# Table of Contents

Executive Summary .................................................................................................................. 3

Introduction ................................................................................................................................. 3

Data Processing ........................................................................................................................... 5
  Key Steps ................................................................................................................................. 5
  Methodology ............................................................................................................................ 5
    Non-Real Time Data Processing .......................................................................................... 5
    Real Time Data Processing ................................................................................................. 10
  Discussion of Methodology and Assumptions ......................................................................... 12

Data Analysis ............................................................................................................................ 13
  Mathematical Formulation ..................................................................................................... 13
  Matlab Model ......................................................................................................................... 15
    Model Sequence for Non-Real Time .................................................................................... 18
    Model Sequence for Real Time .......................................................................................... 18
  Assumptions .......................................................................................................................... 19

Results & Analysis ....................................................................................................................... 19
  Simulation 1 (No new nodes) ................................................................................................. 20
  Simulation 2 (No map updates) ............................................................................................. 21
  Simulation 3 (No difference in variance) ............................................................................ 21
  Simulation 4 (All levers dynamic) ......................................................................................... 22

Conclusions ................................................................................................................................. 22

Next Steps & Recommendations ................................................................................................. 23

Acknowledgements ..................................................................................................................... 24

Works Cited ................................................................................................................................. 25

Appendix A: 24-hour maps from OpenStreetMap ................................................................. 26

Appendix B: Matlab Implementation .......................................................................................... 28
Executive Summary

Our main task was to quantify the value of real time data in disaster settings, specifically urban search and rescue (SAR) teams. To evaluate this, we used the January 2010 earthquake in Haiti as a test case because of the abundance of information available as well as the fact that it was one of the first disasters that utilized crowdsourcing for humanitarian purposes. Our simulation model has two sources of real time data—SMS text messages from 4636 and street map updates from OpenStreetMap. In order to compare the real time case to non-real time, we used a map and classifications of buildings in Port-au-Prince that we were able to extrapolate demand from and use in both cases.

Based on the assumption that real time data reduces the uncertainty in information—both of demand and travel times—we found that real time data is always valuable in terms of people served in SAR. The biggest value seems to come when both types of real time data are incorporated (maps and SMS text messages) and the assumption of reduced variability holds. This implies that field workers in the future should focus on verifying the validity of information, obtaining real time SMS messages with applicable information, and acquiring up to date maps.

Introduction

On January 12, 2010 a violent 7.0 earthquake struck the island of Haiti. More than 230,000 victims were killed by the natural disaster with many others wounded or trapped under rubble. Following the disaster, official responders began the rescue and relief efforts. These responders had to quickly fulfill requests for help, but their efforts were made more difficult with the natural disaster collapsing hospitals and blocking off roads. In order to cope with these changes in geography, responders looked to volunteers to help update the changing terrain. Volunteers at Ushahidi were able to contribute their time and expertise to help the responders in Haiti by updating maps with information gathered through social media such as Twitter, Facebook, and blogs. However, a big challenge arose from sorting and prioritizing the influx of data that the volunteers were receiving, with much of the prioritizing being done on an ad hoc basis.

The purpose of this project is to quantify the benefits of using real time data, specifically text messages, during natural disasters for search and rescue (SAR) teams. In the developing world, cell phone usage has become very widespread. Even in impoverished areas, many people have access to voice and text messaging services. After the 2010 Haiti earthquake, many organizations pulled together a system where people in Haiti could send SMS messages to the number 4636 with their requests for food, aid, etc. and response teams encountered an overwhelming flow of data. Despite outsourcing some of the data to volunteers, the responders did not fully utilize the text messages due to a lack of standard protocol. Many feel that not using these text messages was a lost opportunity to

---

1 Anne Nelson, Ivan Sigal, and Dean Zambrano. "Lessons from Haiti."
2 Morrow, Nathan, Nancy Mock, Adam Papendieck, and Nicholas Kocmich. "Independent Evaluation of the Ushahidi Haiti Project."
3 Heinzelman, Jessica, and Carol Waters. "Crowdsourcing Crisis Information in Disaster- Affected Haiti."
help those affected by the earthquake. Others believe that triaging the immense amount of data would be very difficult to standardize, and bring forth the other issue of validating the incoming text messages. With regard to the latter point, responders to the Haiti earthquake have already shown that volunteer-generated information can be reliable. Since many maps were outdated or nonexistent, official responders to the earthquake often ended up using volunteer-generated maps. This shows promise in utilizing volunteers to sift through text messages and social media in future natural disasters.

This report attempts to quantify the benefits of using real time data so that future natural disaster responders and SAR teams will establish standard protocol for using text messages in the future. As a proof of concept, we have created two models of post-natural disaster urban search and rescue. One of the models represents SAR efforts without real time data while the other model represents search and rescue with real time data. The next section of our report, data processing, discusses our data sources such as Ushahidi for the Haiti earthquake for both the non-real time and real time models. In addition, the data processing section also describes our methodology for processing that data within ArcGIS, a geographic mapping software. The data analysis section of our report will explain our modeling approaches along with the results. We begin with the logical flow of the two models and present our different approaches through using AMPL (a linear programming solver) and MATLAB (a technical computing language). The discussion section of the report will detail the benefits of real time data and the implications for future natural disasters. Finally, the next steps section will offer recommendations to future search and rescue teams and additional work that will further the process of using real time data done in SAR.

---

4 Harvard Humanitarian Initiative. Disaster Relief 2.0:

Data Processing

Key Steps
Data Processing for Non-Real Time Model

1. Obtaining Building Information
   After speaking with Search and Rescue teams, it was understood that before any real time data was available, the teams decided on search locations based on directions from local government and maps. In the case of Haiti, when Search and Rescue teams first arrived on site, with limited information available, they had to make educated guesses on the most probable locations of demand and give priority to buildings with the biggest potential of having the largest number of survivors. Therefore, we decided to estimate demand for the non-real time model based on building location and types, and prioritize them according to estimated demand.

We obtained a Buildings.shp file from Geofabrik, which contained location information and classifications of buildings in Haiti as shown in graph below.

---

7 Geofabrik: OpenStreetMap Extracts for Haiti, 1 Feb. 2012 <http://labs.geofabrik.de/haiti/>
Buildings.shp in ArcGIS

2. Filtering Building File
To identify nodes of interest, we filtered the Buildings.shp file to only include polygons that satisfy the following criteria. After filtering, there were 224 buildings left.

- Located in Port-Au-Prince (used administrative boundaries found on [http://cegrp.cga.harvard.edu/files/haiti/haiti_admin_boundaries.zip](http://cegrp.cga.harvard.edu/files/haiti/haiti_admin_boundaries.zip))
- Is an educational institution (including “school”, “école”, “college”, “kindergarten”), orphanage, shelter or hospital

The reason for selecting those building types is that schools and hospitals are generally densely populated venues and children and patients are more vulnerable than the average population. Therefore, it is our best guess that these places would have high demand. Below a snapshot of the Attribute Table of the Buildings.shp file. None of the buildings shown was included in the set of 224 because the building types did not satisfy our criteria.
3. Calculating Area and Estimating Demand
   Area of buildings was calculated using the Geometry Calculator feature in ArcMap 10 after changing the coordinate system to a projected coordinate system, WGS_1984. To calculate the expected demand at a building, we made the following assumptions:

   - 28,757 persons/Km² (according to the fourth general population and housing census in 2003 by the Institut Haitien de Statistiques et d’Informatique)
   - 10% of residents were trapped

   Therefore, expected demand = Area (m²) * 0.0288 (persons/m²) * (0.1)

4. Prioritizing Demand Nodes
   A smaller set of nodes was needed for the purpose of modeling. Therefore, the demand nodes were trimmed down by selecting the top 133 nodes with the most expected demand.
5. Calculating OD Matrix
Some preparation work needed to be done before generating the origin-destination (OD) matrix in ArcGIS. First, we transformed building polygons into points in ArcMap by locating their centroids. The road information we used came from OpenStreetMap's database for 1/13/2010. We used the file “highway_lines” to map out the roads in Port-au-Prince.

<table>
<thead>
<tr>
<th>OID</th>
<th>osm_id</th>
<th>name</th>
<th>type</th>
<th>X</th>
<th>Y</th>
<th>Absolute_A</th>
<th>Demand</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>48215556</td>
<td>CENTRE HOSPITALIER D</td>
<td>hospital</td>
<td>-72.3258</td>
<td>18.5393</td>
<td>1623.282749</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>48276865</td>
<td>PORT AU PRINCE HOSPITAL</td>
<td>hospital</td>
<td>-72.3455</td>
<td>18.5373</td>
<td>5145.929588</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>48276866</td>
<td>Hôpital St. François de la Paix</td>
<td>hospital</td>
<td>-72.3441</td>
<td>18.5382</td>
<td>12783.853292</td>
<td>31</td>
</tr>
<tr>
<td>3</td>
<td>48285471</td>
<td>Hotel Caraibe Haiti</td>
<td>hotel</td>
<td>-72.3644</td>
<td>18.5308</td>
<td>805.262259</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>48286054</td>
<td>Hôtel Villa Créole</td>
<td>hotel</td>
<td>-72.3292</td>
<td>18.5224</td>
<td>2695.510115</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>48304467</td>
<td>Centre d'Apprentissage</td>
<td>school</td>
<td>-72.3311</td>
<td>18.5533</td>
<td>1258.335792</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>48330852</td>
<td>Gheskio - National HIV</td>
<td>hospital</td>
<td>-72.3489</td>
<td>18.5391</td>
<td>1915.24413</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>48345364</td>
<td>Hôtel Royale Haitien</td>
<td>hotel</td>
<td>-72.3728</td>
<td>18.5314</td>
<td>1225.444667</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>48370677</td>
<td>Ecole Evelyne Levy</td>
<td>building_concret</td>
<td>-72.5382</td>
<td>18.2341</td>
<td>1354.174095</td>
<td>3</td>
</tr>
<tr>
<td>9</td>
<td>48371950</td>
<td>Savanne Christ</td>
<td>building_concret</td>
<td>-72.6926</td>
<td>19.4502</td>
<td>9066.309462</td>
<td>22</td>
</tr>
<tr>
<td>10</td>
<td>48374633</td>
<td>Hospital Frances</td>
<td>hospital</td>
<td>-72.3405</td>
<td>18.5379</td>
<td>1151.332058</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td>48417965</td>
<td>Nursing School (FSIL)</td>
<td>school</td>
<td>-72.6312</td>
<td>18.5151</td>
<td>1512.098967</td>
<td>3</td>
</tr>
<tr>
<td>12</td>
<td>48417966</td>
<td>Nursing School (FSIL)</td>
<td>school</td>
<td>-72.6318</td>
<td>18.5155</td>
<td>1625.56616</td>
<td>4</td>
</tr>
<tr>
<td>13</td>
<td>48423058</td>
<td>École Mixte de Sainte Thérèse</td>
<td>building_concret</td>
<td>-72.3323</td>
<td>18.5772</td>
<td>830.43622</td>
<td>2</td>
</tr>
</tbody>
</table>
To find the time cost of travelling between demand nodes, we further processed the “highway_lines” Shapefile to reflect the difference in accessibility and travel time of different types of roads. In the given data, highways were classified into four categories: primary, secondary, tertiary, and residential. We made assumptions about the travel time for each type of highway and calculated the time cost of travelling along each road.

Assumptions:

<table>
<thead>
<tr>
<th>Highway Type</th>
<th>Time cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary</td>
<td>Length*1</td>
</tr>
<tr>
<td>Secondary</td>
<td>Length*1.2</td>
</tr>
<tr>
<td>Tertiary/Residential/Unclassified</td>
<td>Length*1.4</td>
</tr>
<tr>
<td>Others</td>
<td>Length*1.6</td>
</tr>
</tbody>
</table>

This method takes into account the fact that in practice more time is needed to travel through a residential path than to travel on a major street, and we assumed that it would take 40% more time. Here, the time cost has no units and the numbers are only meaningful relative to each other. When using these numbers in the model, they need to be multiplied by a constant to make them reflect real world circumstances. In addition to varying travel time, we also eliminated from the highway_lines file roads that are inaccessible to vehicles such as “paths”, “pedestrian” and “footway”. The new set of highway lines was saved as “usable_roads.shp”.

A new feature dataset was created in ArcCatalog using “usable_roads.shp”. When creating the Network Dataset, we used the following settings:
• Global turns: Allows turns at any connected intersection in all directions
• No elevation: Assumes a flat land—a simplifying assumption as no elevation data was available
• Any vertex connectivity: An edge may connect to another edge or junction where they have coincident vertices

The feature dataset containing usable road information was used to calculate the OD matrix for the non-real time data.

The stopping time at each node was assumed to be 60 minutes for the first trapped person and 20 minutes for each additional person. Reasoning for this assumption is discussed in Discussion of Methodology and Assumptions.

Real Time Data Processing
1. Filtering Text
The real time demand data used in this project comes from SMS text messages that were sent to 4636 and processed by Ushahidi volunteers. The 32,379 text messages from the raw data set were not all usable for our modeling purposes and needed to be filtered because some contain information that is not relevant to Search and Rescue Teams and some have no geographical coordinates. The fact that Ushahidi has assigned each text into a specific category made it easy to do filtering on content. By performing simple queries in Excel, we picked out texts that had coordinates and were in one of the following categories:

- “People Trapped”
- “Person Trapped”
- “Collapsed Structure”
- “Person News”
- “Missing Person”

For the purpose of modeling, we chose to work with the first five days of relevant text messages, i.e. 1/17/2010 – 1/21/2010. After filtering with the aforementioned criteria, we were left with 22 relevant texts, which we will refer to as demand nodes. The following graph shows the text table highlighting texts that we used as demand nodes.
2. Obtaining Geo Coordinates

All 22 SMS texts have both original longitude and latitude coordinates as well as fields called “ush_long” and “ush_lat”. We used the latter set for mapping because their coordinates were verified by Ushahidi volunteers and provide more consistency.

3. Building Maps

Since there was very limited data available prior to the earthquake, volunteers updated the maps of Haiti constantly to provide more accurate information to teams in the field. To fully utilize real time information available on the cloud, we utilized OpenStreetMap (OSM) data to develop a new map for each day to reflect changes in road condition. First, we started with a map of the world on 1/13/2010 found on OpenStreetMap and extracted only the portion of Port-au-Prince which will be used as the base map. Then, we applied daily change sets—a group of edits within 24 hours—found on OSM to the base map to reflect the most updated road conditions for each day following the earthquake. Change sets were applied to the base map using Osmosis which is a command line Java application for processing OSM data. As a result, we had a new Geo Database for each day from 1/17/2010 to 1/21/2010, a period that corresponds to the incoming of relevant texts.

4. Mapping Demand Nodes

For each day after the earthquake, a new map is generated and new demand nodes are added to the map in ArcGIS. Since relevant texts started coming in on 1/17/2010, that day is referred to as Day 1, and so forth. To build a mapping for Day 1, the “highway_line” Shapfile from Day 1’s Geo Database and demand nodes from Day 1 were added as layers to ArcMap. For example, there were two relevant text messages on Day 1 and therefore only...
two real time demand nodes, together with the 133 non-real time nodes, there are 135 nodes in total. The nodes were mapped using “ush_long” and “ush_lat” coordinates.

5. Calculating OD Matrix via Network Analysis
We used the same method as the non-real time network analysis, using an updated map for each day to build a new Feature Dataset. Below is an overview of the size of calculations for each day.

<table>
<thead>
<tr>
<th>Day</th>
<th>Number of Non-Real Time Nodes</th>
<th>Number of New Nodes</th>
<th>Total Number of Nodes</th>
<th>Size of OD Matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>133</td>
<td></td>
<td>133*133</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>133</td>
<td>2</td>
<td>133+2=135</td>
<td>135*135</td>
</tr>
<tr>
<td>2</td>
<td>133</td>
<td>9</td>
<td>135+9=144</td>
<td>144*144</td>
</tr>
<tr>
<td>3</td>
<td>133</td>
<td>4</td>
<td>144+4=148</td>
<td>148*148</td>
</tr>
<tr>
<td>4</td>
<td>133</td>
<td>4</td>
<td>152</td>
<td>152*152</td>
</tr>
<tr>
<td>5</td>
<td>133</td>
<td>3</td>
<td>155</td>
<td>155*155</td>
</tr>
</tbody>
</table>

Each of the six matrices described above was used to solve a routing problem using the model described later in the report.

Discussion of Methodology and Assumptions
1. Demand Estimation
The estimation for demand of the non-real time model is solely based on area of buildings, which is a simplifying assumption. We did not take into account the elevation of the buildings, which could be significant if the building is a high rise. However, Port-au-Prince has very few high rises and thus this assumption would not bring major flaws to the model. In addition, we used population density figures from the 2003 census data which is a little outdated considering that the earthquake took place in 2010. An alternative is to use the 2009 estimated population density figure, which has more variance, as it is a projection.

2. Selection of Demand Nodes for Non-Real Time Model
Out of the 3000 building polygons, we picked 133 to be demand nodes, which were mainly schools and hospitals, for the model. Although schools and hospitals are generally densely populated areas and thus have a higher chance of having trapped persons, we acknowledge that there exist other building types such as office or residence, which may also have high numbers of trapped persons. More building information such as number of tenants, elevation, and structural material would help us to make better judgment on selecting top demand nodes, which are the most vulnerable to earthquake and would have more trapped persons. However, this data was not available at the time of the earthquake so we did not include it in order to model what SAR teams would have encountered when they arrived.
3. Time Cost Estimation
The time cost estimations reflect the difference in road types under consideration. The actual numbers assigned were somewhat arbitrary. For example, there is no real evidence that a highway labeled “secondary” would take 20% more time to travel through than a highway labeled “primary”. Therefore, although our time cost estimation do account for the different of road types, it is subject to systematic errors. To mitigate these errors, we would need to know the precise definition and meaning of the classifications.

4. Estimation of Stopping Time
We assumed that the first person would take 60 minutes to rescue and each additional person after that would take 20 minutes. This is a simplifying assumption that we needed to make in order to run the model, but we do acknowledge its shortcomings because of the wide range of wait time possibilities. For example, sometimes a severely trapped person may need to be amputated on the site in order to be pulled out from the rubbles, a process that can take up to 6 hours; other times, a person only need to be carried onto a car, which only takes a few minutes. As it is very hard to predict the scenarios at each site, we used a mean time of 60 minutes for the first person to account for the time needed to come up with a rescue plan and do any preparation work that is needed for the rescue. We used a mean time of 20 minutes for each additional person that follows since the preparation work is already done and the additional time only accounts for the rescue portion.

5. Filtration of Text Data
Though there were over 30,000 texts received through 4636 after the earthquake, the majority was not relevant to Search and Rescue teams which is why we ended up with only 22 texts, which dated from 1/17/2010 to 1/21/2010. We may have missed nodes because of the inaccuracy in translation or categorization by Ushahidi volunteers. Furthermore, the drawback of SMS texts is that they do not contain demand information—how many people are trapped—and a lot of them did not contain location information. Those in charge of 4636 should instruct people to give location information and an estimate of how many people are trapped at the location so to better facilitate Search and Rescue teams’ decision making.

Data Analysis
After determining that the scope of our project was to evaluate the benefits of taking into account real time data, our initial decision, which we carried through our project, was to compare this by looking at lives saved or “demand serviced”. We started off by localizing the problem and creating a scaled down model with minimal data inputs to be solved in AMPL. The simplified model solves a problem involving one vehicle that wants to maximize demand in a given amount of time.

Mathematical Formulation
Index:
0: Depot designation
i, j: Node designation

Parameters and Constants:
Di = Demand at node i
Cij = Travel time from node i to j
Si = Stopping time for node i
T = Total time available for period n

Variables:
Xij = Binary, 1 if van travels from node i to j, 0 otherwise
Yij = Binary, 1 if van k visits node i, 0 otherwise

Objective Function:
Maximize:
\[ \sum_l Y_l \cdot D_l \]
Subject to:
\[ \sum_j \sum_i X_{ij} \cdot C_{ij} + Y_l \cdot S_l \leq T \]
\[ \sum_j X_{ij} = \sum_j X_{ji} \]
\[ \sum_j X_{ij} = Y_l \]

Our objective function sought to maximize demand serviced subject to limited time. After feeding in a small dataset, this model was able to solve for an optimal solution. Although the AMPL model was able to efficiently solve the basic problem, it was unable to accommodate the following intricacies:

- For both the real time and non-real time model, we required a constantly updating model over a course of time (5 days). AMPL would only be able to solve for one day at a time and we would need to manually update all aspects of the model after each day, i.e. removing nodes visited and updating with real time OD matrices.
• The premise of our model is based on the concept of variance and how real time data allows us to assume a smaller variance in demand, stopping time, and travel time. AMPL would require these data sets to be generated ahead of time to be fed into the model. Since we planned on running this simulation at least 50 times, creating 250 (50 * 5 days) data sets would be highly inefficient.

• In TSP modeling, subtour constraints are needed to ensure that all nodes are visited in one single trip with a single starting point and returning back to that starting point. In this model, the constraint that every node must be visited was relaxed since the objective is to maximize demand from nodes in a given amount of time rather than visiting all of them. For example, without subtour constraints if nodes 1, 2, and 3 were chosen, we could end up with a path like the diagram below where each node is visited and the vehicle ends back at the base node, but there’s no connector among the two segments:

With the inclusion of subtour constraints, we avoid these two node loops. However, the number of constraints needed is $2^n$ where $n$ is the number of nodes. With a mere 10 nodes (significantly less than reality) we would already require 1024 subtour constraints.

**Matlab Model**

We decided to code the model in Matlab after realizing it could handle the majority of our specifications. The main downside to Matlab was the lack of a robust optimization function; however we developed a heuristic to solve for the best solution with a very small solution time. Below is a flow diagram of the model and the steps that it takes to reach the efficient output:
Inputs:

- **Origin Destination (OD) Matrix**: Distances between the nodes
  - Day 0 matrix is used throughout the non-real time model
  - Real time model matrix is updated daily with a new layer of additional information received from texts as well as a new map from the change sets
Above is a truncated image of what the OD matrix looks like. As labeled, the first two columns are the nodes being considered, the third is the travel time between the nodes, and the last column pertains to period this information applies to. For non-real time information the only portion of the OD matrix considered is period 0 whereas for real time information there is a new matrix for each period.

- **Demand Stop Day (DSD) Matrix:** Demand at each node
  - Similar to above, the DSD Matrix for the non-real time model is constant while the DSD matrix for the real time model is updated daily with any new incoming information.

<table>
<thead>
<tr>
<th>Node A</th>
<th>Node B</th>
<th>Travel Time</th>
<th>Period</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>11</td>
<td>0.026814</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>0.001821</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>0.001821</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>28</td>
<td>0.023191</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>0.00196</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>0.00196</td>
<td>1</td>
</tr>
</tbody>
</table>

Above is a truncated image of the DSD matrix. The first column corresponds to the node, the second is the demand at the node, next is the amount of time spent servicing the demand (function of demand), the fourth column is the period this information is available (non-real time simulation only considers period 0 nodes which are all part of the base map while new nodes that become available are used in the real time simulation), lastly the remaining two columns are the longitude and latitude coordinates for the node.

- **Periods:** Number of days to run our simulation
- **Daily Time:** Constraint that limits the amount of time in a day depending on the work schedule of the search and rescue teams
- **Variance:** Set as a constant percentage of the mean of the inputs. Non-real time data has a higher variance than real time data due to less information available and therefore higher uncertainty.
Outputs:

- Total demand covered over n periods including a breakdown of demand covered in each period

**Model Sequence for Non-Real Time**

1. All the inputs are recorded in csv files which are fed into Matlab
2. Variance is set at a higher point than that for real time data
3. Origin node (location where the team returns to at the end of each day and starts off every morning) is chosen as the node with the greatest ratio of (demand)/(total distance to other nodes)
4. Sort remaining nodes by the same ratio of (demand)/(total distance to other selected nodes)
5. Choose the highest node after the sorting (other than the original node) and add to the group of nodes that will be visited that period
6. The demand of the added node subject to variance is included in total demand
7. A new distance matrix is created consisting of the newly added node as well as all previous nodes
8. This set of nodes along with the new distance matrix is entered into the Traveling Salesman Problem (TSP) genetic algorithm which solves for the optimal route (travel from the original node through all the nodes and return back to the original node in the shortest time)
9. The travel time and stopping time (a function of demand at each node) are both subject to variance and the new total time is the sum of all the travel times and stopping times
10. The nodes visited are removed from consideration
11. If the time it takes to visit all the nodes is less than the total time in a period, the model will go back through the list of nodes ordered by (demand)/(total distance to other selected nodes) and pick the next highest one
12. This process is repeated for each period until the last one or until there are no more nodes to service

**Model Sequence for Real Time**

1. All the inputs are recorded on csv files which are fed into Matlab
2. Variance is set at a lower point than that for non-real time data
3. Origin node (location where the team returns to at the end of each day and starts off every morning) is chosen as the node with the greatest ratio of (demand)/(total distance to other nodes)
4. Sort remaining nodes by the same ratio of (demand)/(total distance to other selected nodes)
5. Choose the highest node after the sorting (other than the original node) and add to the group of nodes that will be visited that period
6. The demand of the added node subject to variance is included in total demand
7. A new distance matrix is created consisting of the newly added node as well as all previous nodes

---

10 A search heuristic that mimics the process of natural evolution. This heuristic is routinely used to generate useful solutions to optimization and search problems
8. This set of nodes along with the new distance matrix is entered into the Traveling Salesman Problem (TSP) genetic algorithm which solves for the optimal route (travel from the original node through all the nodes and return back to the original node in the shortest time).

9. The travel time and stopping time (a function of demand at each node) are both subject to variance and the new total time is the sum of all the travel times and stopping times.

10. The nodes visited are removed from consideration.

11. If the time it takes to visit all the nodes is less than the total time in a period, the model will go back through the list of nodes ordered by (demand)/(total distance to other selected nodes) and pick the next highest one.

12. During the next period, the inputs are updated with any new information that has come in. For example, the OD matrix could be updated with an additional node or changes to the time it takes to get between two nodes due to the updated map. After the inputs are updated, steps 4 through 11 are run again until the last period or until all the nodes have been serviced (note however in the case of real time even if all existing nodes are serviced by an earlier period, it will still run the later periods since additional nodes could come up with new information).

Assumptions

- Each additional text or information regarding a node decreases variability, i.e. the information is reliable (demand, stop time (function of demand), time it takes to get from node to node). In reality, the validity of the information is a lot more variable and there’s a much higher chance that the information received could lead nowhere.

- Set base node according to greatest ratio of demand to total distance to other nodes and return to it every night/start there every morning. In reality, a lot of other factors will go into setting the base node such as proximity to transportation hubs where workers arrive.

- Base map (OD Matrix and DSD Matrix) is stagnant for non-real time.

- Base map is layered with additional information/nodes for real time, i.e. updated OD Matrix and DSD Matrix each period.

- Variance is set as greater for non-real time than for real time and is set at a constant percentage of the mean (20% for non-real time and 10% for real time).

- As data is coming in each day, nodes can only be uncovered on days the information is received.

- Service time, or time spent at each node rescuing people is linear. In reality it is likely that a certain percentage will need more time and attention while the others require less time.

- Focusing on one vehicle or one search and rescue team.

- Ignoring the “spoke” model which is more likely to be the case in real life since the individuals rescued will need to be taken back to base for medical attention.

Results & Analysis

In order to best compare the value add of real time information, we ran the simulation under four different conditions:
The X’s indicate the levers of the simulation we held constant for real time (for example in simulation 1 we didn’t allow for any new nodes in real time information, only updated maps and a different variance) while the √’s indicate levers we kept dynamic. Each simulation was run for 50 iterations and results shown below are in box plots where each pair of plots corresponds to the comparison of real time and non-real time demand spreads for each period. The numbers listed under each box denotes the average demand for that day over 50 iterations:

**Simulation 1 (No new nodes)**

<table>
<thead>
<tr>
<th></th>
<th>New Nodes</th>
<th>Map Changes</th>
<th>Different Variance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation 1</td>
<td>X</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Simulation 2</td>
<td>√</td>
<td>X</td>
<td>√</td>
</tr>
<tr>
<td>Simulation 3</td>
<td>√</td>
<td>√</td>
<td>X</td>
</tr>
<tr>
<td>Simulation 4</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
</tbody>
</table>

While holding the number of nodes constant (excluding SMS texts from 4636) and including map updates (increase or decrease in travel time) as well as a different variance for real time over most of the periods with the exclusion of period 5, real time data results in a higher average demand serviced. The variance for the real time demand simulation is also consistently lower which is what we expect given settings that non-real time has higher uncertainty. Since there are no incoming nodes, the average demand serviced for each period is similar, whereas we’ll see in later simulations that there is a huge spike in day 4 due to additional nodes being included. The heuristic targets the most efficient nodes (those with the highest demand/total distance to all selected nodes ratio) in the beginning and therefore have the highest averages in the first day and tapers off as the remaining nodes have lower demands.
Simulation 2 (No map updates)

For this simulation, no new information regarding changes in the map were introduced, therefore both real time and non-real time were subject to the same base OD matrix over the 5 periods. Results here are on par as we see that the average demand serviced for each period is consistently higher for the real time simulation and variance is narrower. One particular fact to note is the large discrepancy in serviced demand in day 4, which was not evident in simulation 1 due to a large influx of demand from newly uncovered nodes.

Simulation 3 (No difference in variance)
We see clearly in this result that the variance of the demand serviced across the periods is much closer for both real time and non-real, which is what we expected, given a constant variance. Again, we see that average demand serviced is consistently higher for real time than for non-real time. Similar to simulation 2, we continue to observe the strong effect that including additional nodes has on demand serviced for later days.

**Simulation 4 (All levers dynamic)**

For the last simulation, all levers were fair game and we observed the combined effects of each of them. As can be seen in the graph above, real time information clearly adds value in terms of people saved.

**Conclusions**

From the aggregate run as well as each of the results from before where one variable was held constant, we extrapolated the following conclusions:

- Without the inclusion of new nodes via information from text messages, the pattern of the average demand serviced is the same for real time and non-real time since heavy demand nodes are targeted in the beginning and as those candidates are removed it became a matter of which would get through everything the fastest. When node updates were included we see a noticeable increase in day 4 from the incoming information detailing locations of additional nodes and potential demand.

- By excluding map update information, there is a subtler factor removed. Real time demand serviced still consistently outweighs non-real time due to the other two levers. However, by not including information detailing changes to road conditions and travel times, the pool of selected nodes is different and the optimal set is not necessarily chosen.
• Setting different variances for real and non-real time information has a significant role in the results. In simulation 3 we see a much more uniform level of variance over the 5 periods between non-real time and real time while every other simulation which included the discrepancy in variance indicated a much larger spread for non-real time demand serviced in comparison to real time demand.

Next Steps & Recommendations

We have several recommendations for future disaster relief teams. For urban SAR teams, the most valuable real time data is that relating to building damages or collapses, road damages or blockages, and missing or trapped people. One caveat to this is that the incoming information is only useful to SAR teams if location information is included or can be extracted from the SMS texts. We would therefore recommend that future systems like 4636 encourage people that are sending the messages to include this information. Also, future systems may be able to use geolocation tools in newer cellular devices to pinpoint the exact location of each incoming SMS message.

Although we believe our current heuristic is accurate enough to examine the value add of real time data, the following is a list of action items we recommend as next steps for this project:

• Our current heuristic implements a TSP genetic algorithm, which is has characteristics similar to the problem we are solving, but a lot of major differences as well. A more realistic model would be a dynamic vehicle routing problem.\(^{11}\)

• The approach should be integrated so that although each team is responsible for their own regions, and optimization of assignments should take into account the entire map and each team’s relative location. For example, although a certain node is within team A’s region, they could be further away from it than team B.

• Future research teams should also look into including more restrictions and remove more of the assumptions. For example, an actual base node should be set rather than one based off of the highest ratio.

• A more realistic method to represent the validity of data is also very important. The current model makes the assumption that each additional piece of information is beneficial and reduces the uncertainty or variance of the input information. However, this is not the case in real life. Often times faulty information comes through that can throw off the SAR teams if they receive false leads.

• The stop/service time at each node should be differentiated such that different groups of individuals have different time needs. For example x% might need A hours, y% needs B hours, and z% needs C hours where x + y + z = 100%.

• There is a caveat in the current model in the addition of nodes within a certain period. The function keeps running within a certain period and adding the next node with the highest \(\frac{(demand)}{(total\ distance\ to\ each\ selected\ node)}\) to the group to visit until the time constraint is met. Even if the travel time + stop time to add the next highest node exceeds the time

\(^{11}\) The vehicle routing problem examines how to service a number of customers given a fleet of vehicles. Our current problem hones in on a particular region and one vehicle, but in the case with multiple regions and vehicles it would be inefficient to solve each separately.
remaining in the period it is still included. This simplification should be corrected for in future models to add the next most valuable node that is within the remaining time.

- The model should take into account “spoke” traveling where either the SAR team or a portion of it must return back to the base node after visiting certain nodes where individuals require medical attention that can only be provided for at the base.

Acknowledgements

There are many individuals we would like to thank for lending us their expertise throughout this project. Without their guidance, much of this project would not be possible. We would like to thank Dr. Jennifer Chan for helping us understand the field of search and rescue as well as for giving us this opportunity to leverage our skills for humanitarian efforts. We would like to thank Professor Irina Dolinskaya for meeting with us weekly and encouraging us to work through this complex project. We would also like to thank Edwin Shi for his math modeling expertise and his advice on improving our model. We would like to thank Lewis Meineke for teaching us an ArcGIS course as well as fielding our numerous mapping questions. Finally we would like to thank Professor Mark Werwath for organizing the design course and giving us the opportunity to demonstrate our Industrial Engineering skills.
Works Cited


Appendix A: 24-hour maps from OpenStreetMap

Map from OSM of Port-au-Prince on 1/13/2010

Map from OSM of Port-au-Prince on 1/17/2010
Map from OSM of Port-au-Prince on 1/18/2010

Map from OSM of Port-au-Prince on 1/19/2010
Appendix B: Matlab Implementation

Masterfunction.m

%Primary function for accepting inputs of OD, demands/stops for nodes, timecost information, # periods, realtime or not, variance weight and variance multiplier. Will return the final demand covered in n periods for each period as well as a matrix for each period detailing the chosen coverage nodes and optimal route.

%See parameters.m for example parameters used and simulation of this function.

function varargout = masterfunction(od,dsday,dailytime,timecost,periods,realtime,varweight,varmultiplier)

precount = length(dsday(:,1,1));

%genetic algorithm parameters
popSize = 60;
numIter = 1e3;
showProg = 0;
showResult = 0;

%adjust model for realtime or non real time
if realtime
    %adjust variance for realtime
    varianceweight=varweight;

    %selects baseod distances only for days beyond 0
    baseod(:,:) = od(od(:,4)>=0,:);

    % Remove after running - Testing real time model under no OSM data
    %baseod(:,:) = od(od(:,4)==5,:);
    %clear od;
    %od(:,:) = baseod(baseod(:,4)==5,:);
    % Remove after running

    % Remove after running - Testing real time model under no 4636 data
    %dsday = dsday((dsday(:,4)==0),:);
    % Remove after running

else
    %selects baseod distances only for day 0
    dsday = dsday((dsday(:,4)==0),:);

    %adjust variance for nonrealtime
    varianceweight=varweight*varmultiplier;
    %selects baseod distances only for day 1

end
baseod(:,:,1)=od(od(:,4)==1,:);
clear od;
od(:,:,1)=baseod;
end

%optRoute = zeros(length(dsday(:,1,1)),length(dsday(:,1,1)),periods);

%set totaldemand and totaltime for first round to be zero as at origin
%depot
totaldemand(1,:) = [repmat(0,periods,1)'];
totaltime(1,:) = [repmat(0,periods,1)'];

%p for loop is a period/day
for p=1:periods;
    i(p)=1;
    %Only considers nodes available on day <= p
    basedsday=dsday((dsday(:,4)<=p),:);

    % Keep after running - Testing real time model under no OSM data
    if realtime
        clear od;
        %Switch to basemap for day p
        od(:,:,1) = baseod(baseod(:,4)==p,:);
    end
    % Keep after running - Comment out switching basemaps, keep this
    % part when testing no OSM data.

    %presort nodes by demand distance weight.
    %for first period only, adds first node as origin to start off each
    %round. it is chosen as the node with the greatest demand divided
    %by the total distance to all other nodes.
    if p == 1
        dsort(1:length(basedsday),:,p) = sortrows(basedsday,-2);
        ddsort(:,:,p)=[dsort(:,1:6,p),repmat(0,length(dsort),1),repmat(0,length(dsort ),1)];
        for ii=1:length(ddsor)'
            dsort(ii,7,p) = (sum(od((od(:,1)==ddsor(ii,1,p)),3))*timecost + dsort(ii,3,p))/10000;
            dsort(ii,8,p) = ddsor(ii,2,p)/ddsort(ii,7,p)*1000;
        end
        dsort = sortrows(ddsor,-8);
        stoptime(i(p),p) = abs(normrnd(dsort(i(p),3,p),dsort(i(p),3,p)*varianceweight));

        %Once it is chose, the
        %first totaltime and totaldemand are set to that node.
        totaltime(i(p),p) = stoptime(i(p),p);
        totaldemand(i(p),p) = abs(normrnd(dsort(i(p),2,p),dsort(i(p),2,p)*varianceweight));
%First origin node is entered into considered nodes and has %demand set to 0 to prevent future selection
nodes(i(p),1,1:periods) = dsort(i(p),1,p);
dsort(1,2,p)=0;
else
%For subsequent periods, the demand sorts must remove %previously selected nodes
dsort(1:length(basedsday),1:6,p) = sortrows(basedsday,-2);
for im=0:p-2
    for in=1:nnz(nodes(i(p-1-im),:,p-1-im))
        dsort(find(dsort(:,1,p)==nodes(i(p-1-im),in,p-1-im)),1) ,2 ,p)=0;
    end
end
%First round in each period is set as the origin and thus no %stopping time or demand is captured.
dsort(:,:,p)=sortrows(dsort(:,:,p),-2);
stoptime(i(p),p) = 0;
totaltime(i(p),p) = stoptime(i(p),p);
totaldemand(i(p),p) = 0;
end
minDist(i(p),p) = 0;

%i for loop adds one more node each iteration to tsp genetic algorithm
while totaltime(i(p),p) < dailytime && i(p) < length(dsort(:,1,p)) &&
dsort(i(p)+1,2,p) > 0
    %set demand of visited node to 0
    %add one more node
    i(p) = i(p)+1;
    %sorting the nodes by greatest demand over distance to current %nodes being visited
    ddsort(1:length(dsort),:,p)=[dsort(:,1:6,p),repmat(0,length(dsort),1),repmat(0,length(dsort),1)];
    for ii=1:length(ddsort)
        distsum=0;
        for iii=1:nnz(nodes(i(p)-1,:,p))
            distsum = distsum + od((od(:,1)==ddsort(ii,1,p) &
                od(:,2)==nodes(i(p)-1,iii,p)),3);
        end
        ddsort(ii,7,p) = distsum*timecost + dsort(ii,3,p)/1000;
        ddsort(ii,8,p) = ddsort(ii,2,p)/ddsort(ii,7,p)*1000;
    end
    dsort(:,:,p) = sortrows(ddsort(:,:,p),-8);
    %add the stop time randomly generated
    stoptime(i(p),p) = stoptime(i(p)-1,p) +
        abs(normrnd(dsort(1,3,p),dsort(1,3,p)*varianceweight));
nodes(i(p),1:i(p),p) = [nodes(i(p)-1,1:nnz(nodes(i(p)-
1,:,p)),p),dsort(1,1,p)];

%xy = 10*rand(i(p),2);

%calculating a distance matrix from the od for the nodes in
%consideration
for ix=1:i(p)
    for iy=1:i(p)

        %{ 
        if (isempty(od(find(od(:,1)==nodes(ix) & od(:,2) ==
nodes(iy)),3)))
            x1=basecoords(find(basecoords(:,1)==nodes(ix)),2);
            y1=basecoords(find(basecoords(:,1)==nodes(ix)),3);
            x2=basecoords(find(basecoords(:,1)==nodes(iy)),2);
            y2=basecoords(find(basecoords(:,1)==nodes(iy)),3);
            dmat(ix,iy,p)= varianceweight*sqrt((x1-x2)^2+(y1-
y2)^2);
        else
            dmat(ix,iy,p)=od((od(:,1)==nodes(i(p),ix,p) & od(:,2)
== nodes(i(p),iy,p)),3); 
        end

        end
    end

end

clear xy;

%calculating an xy coordinate set for the nodes
for ix=1:i(p)
    xy(ix,:) = dsort((dsort(:,1,p)==(nodes(i(p),ix,p))),5:6);
end

%using the nodes in consideration, dmat and xy coordinates in
%the genetic algorithm to produce the optimal route and minimum
%time to span these nodes
[optRoute(i(p),1:i(p),p),minDist(i(p),p)] =
tsp_ga(xy,dmat(1:i(p),1:i(p),p),popSize,numIter,showProg,showResult);

%update the totaltime and distance covered with these nodes
totaltime(i(p),p) = stoptime(i(p),p) +
abs(normrnd(timecost*minDist(i(p),p),timecost*minDist(i(p),p)*varianceweight) )

totaldemand(i(p),p) = totaldemand(i(p)-1,p) +
abs(normrnd(dsort(1,2,p),dsort(1,2,p)*varianceweight));

dsort(1,2,p)=0;
end

%set the final output
definaldemand(p) = totaldemand(i(p),p);
end

%set the function outputs
varargout{1} = finaldemand;
varargout{2} = nodes;
varargout{3} = optRoute;
end

parameters.m

%Example parameters provided to masterfunction for Haiti search and rescue analysis. This function expects an 'od.csv' file to contain the origin destination pair distances for each node. Also expected is a 'demandstopday.csv' to contain the demand available, stopping time, day of availability, and coordinates of each node.
clear;
clc;

%load in od and demand/stop data
od=csvread('od.csv');
dsday=csvread('demandstopday.csv');

dailytime = 10000;
timecost = 8080;

varweight = .1;
varmultiplier = 2;

% Remove after running - Testing real time model under same variance
% varmultiplier = 1;
% Remove after running
% number of periods to run the model, should only be set up to 5 for sample
% data as 5 days of OSM maps and ushahidi data are provided.
periods = 5;

% Simulation
for s=1:1
    % basic output to gauge how long the model is taking to run, can be
    % disabled
    c(s,:)=clock

    % running the model once under real time assumptions
    realtime=1;
    [rtfinaldemand(s,:),rtnodes,rtoptRoute] = masterfunction(od,dsday,dailytime,timecost,periods,realtime,varweight,varmultiplier);
    realtime=0;

    % running the model once under non real time assumptions
    [finaldemand(s,:),nodes,optRoute] = masterfunction(od,dsday,dailytime,timecost,periods,realtime,varweight,varmultiplier);

    % node and optimal route outputs are cleared, but can be retained if
    % desired
    clear rtnodes;
    clear rtoptRoute;
    clear nodes;
    clear optRoute;
end