Efficient Evacuation of At-Risk Populations

New Mexico Supercomputing Challenge

Final Report

April 4th, 2017

Team Number: SFHS 162

Santa Fe High School

Team Members:

Rowan Cahill

Ramona Park

Luke Shankin

Teacher Sponsor:

Brian Smith

Teacher Sponsor and Mentor:

Hope Cahill

Table of Contents

Table of Figures

Executive Summary

In light of the numerous natural and man-made disasters affecting the United States, if and how a population evacuates have become pertinent questions. For instance, during Hurricane Harvey, the city of Houston questioned and chose not to issue mandatory evacuation orders due to their experience with Hurricane Rita. On the other hand, in Florida and Puerto Rico such orders were given out and, in the case of the former, an organized method was used. The goal of our simulation was to determine which real-world method of evacuation should be utilized via observation of the time taken for complete egress. These methods include contraflow, chaotic (or unplanned), and segmented evacuation.

Over the course of this project, research concerning real world evacuation plans and statistics from past evacuations was conducted. A controlled two-dimensional Netlogo model was developed from this research. The model does not include unpredictable or complex human behaviors and is therefore limited in its applicability and accuracy. However, it provides a baseline for a more in-depth simulation. Preliminary data from the simulation indicates there is little time difference between contraflow and chaotic evacuation. On the other hand, segmented evacuation results in a greater time duration for complete egress. The length of time recorded for this was roughly proportional to the number of segments evacuated in relation to chaotic evacuation.

Introduction

Purpose

Given recent events, including natural disasters and terrorist attacks within populated areas, the questions of what the most effective method of evacuation is and whether there is a need to evacuate have become increasingly relevant. In response, many cities and areas commonly affected by threats requiring evacuation and even those less threatened are in the process of developing egress strategies. Unfortunately, as is the case with Hurricane Rita, history has demonstrated that the methods proposed don't always work and can at times cause more of a risk then staying put.

To determine which method results in the fastest evacuation of a region our code simulates several of these methods within a grid-like road system. These include contraflow, chaotic, and segmented evacuation. In developing this rudimentary model, multiple algorithms have been tested to provide a baseline for future applications. With modification, the model created could be utilized by city planners to test potential methods of evacuation before a disaster occurs.

Significance

Over the past several years, sudden threats to communities have become an imminent problem both locally and worldwide. These threats, such as hurricanes and wildfires, often require a planned response, including threat assessment and evacuation.

Many recent headlines highlight this issue. For instance, when Houston, Texas, the nation's fourth largest city, was struck by Hurricane Harvey in 2017 government officials chose not to evacuate 2.3 million residents based on flawed procedures during Hurricane Rita in September 2005. Their likely reasoning, "of the more than 100 people who died during Hurricane Rita, at least 60 of those deaths were in connection to the evacuation [1].” Moreover, as reported by the BBC, flawed instructions caused “millions of Texans to flee their homes and led to 20-hour traffic jams, stranding drivers across hundreds of miles [2].” 108 casualties have since been attributed to Hurricane Harvey. It is impossible to say whether these deaths could have been avoided by evacuation.

Figure 1: Death toll rises as flood subsides in Texas
Source: *MSNBC/NBC News*

Figure 2: Traffic jams on highways in Miami, Florida
Source: *zmescience*

Similarly, Hurricane Irma, which traveled across the Caribbean Islands, Cuba, and Florida, resulted in the evacuation of upwards of 6.3 million people—one of the largest evacuations seen in U.S. history [3]. According to the Washington Post, however, Florida's two main roads, Interstates 95 and 75, became "parking lots." The roads reportedly had no exits for "miles at a time" and families were faced with the option of ignoring government orders and staying in their homes, or risk running out of gas before being able to get off the highway.

More recently, in Southern California, fire officials ordered evacuations in response to the Thomas Fire, resulting in tens of thousands of residents fleeing their homes. Despite casualties being less significant than those seen in the aforementioned situations, the devastation to California was undeniably brutal. More than 259,000 acres were burned and over 1,000 structures were destroyed with about 18,000 more homes at risk of being burned [4]. It is difficult to imagine, in this instance, that the evacuation orders issued didn't save lives.

This study is significant on a local level as areas in New Mexico, such as the Sandia and Los Alamos National laboratories, are at risk of nuclear leakage or terrorist and/or military attacks. Additionally, the potential of forest fires, like those in California, is undeniable for New Mexicans, as evidenced by the 2017 wild-fire season. Any of these hypothetical situations could require evacuation of the local populace and fires have.

Background

Evacuations fall into two overarching categories, small-scale and regional evacuation. By definition, a small-scale evacuation is localized; they are often the result of a sudden, violent occurrence [5]. The evacuations implemented after the attacks on the World Trade Center and the Boston Marathon Bombing are an example of this. In these scenarios, while it was necessary to evacuate neighborhoods, it was unnecessary to evacuate the entire city or region.

Figure 3: 9/11 al Qaeda Attack on World Trade Center leaves lasting effects on NYPD in present day
Source: APA Photo/NYPD via ABC New

In contrast, a regional evacuation is instigated by a large-scale threat. Natural disasters, including hurricanes or tsunamis, that have the potential to cause wide scale damage and pose a large casualty risk qualify as such. Evacuation of all areas affected may be deemed necessary during these events [5].

Models of evacuation are further subdivided into maximum flow (MFM) and minimum-cost maximum flow (MC-MFM) models. In both cases, the most efficient mode of egress, that which takes the least amount of time to evacuate the maximum number of subjects, is desired. A key difference is that the MC-MFM factors in the possible risks and costs associated with evacuationas its intent is to combine efficiency with minimal loss of life and costs [5]. The latter are not considered in the development of MFMs.

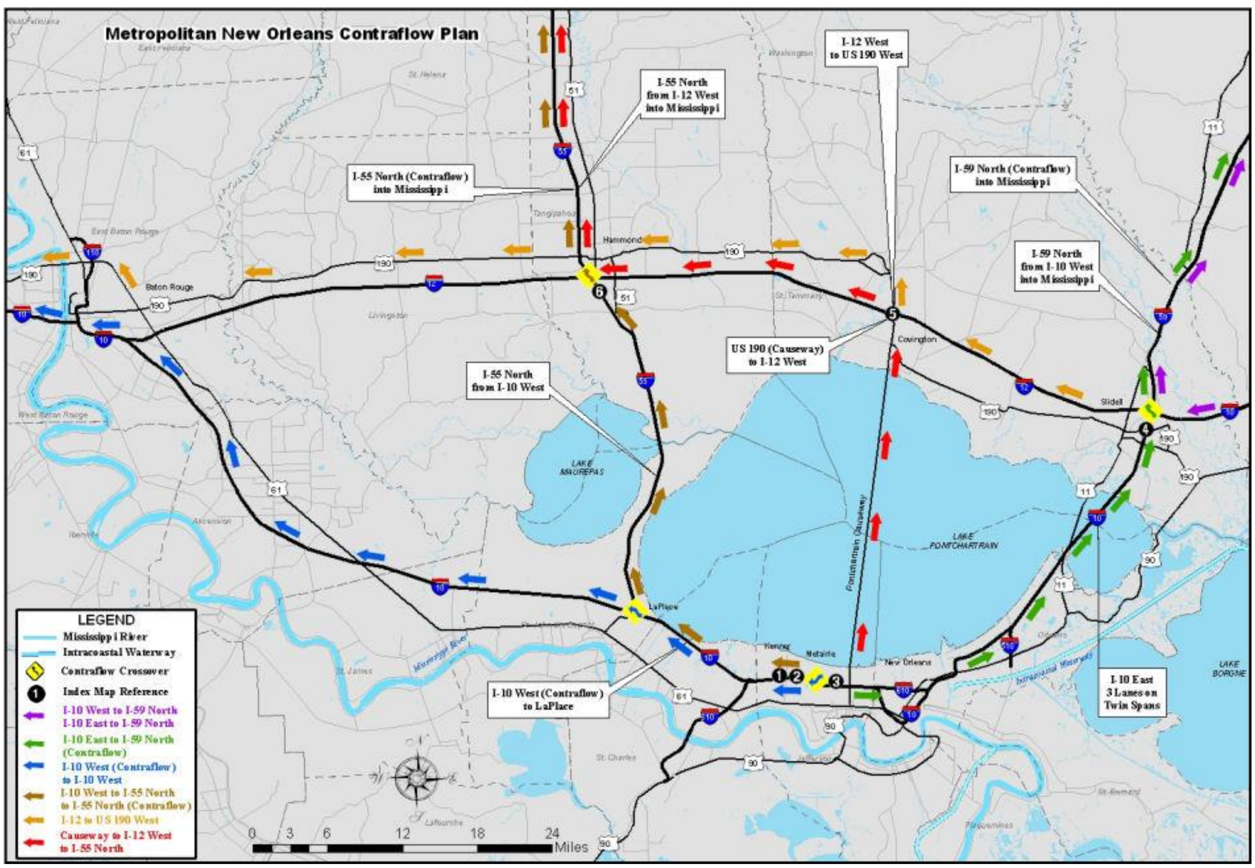

Figure 4: City of Kenner, Louisiana central flow map hurricane information. Source: Louisiana Citizen Awareness and Disaster Evacuation Guide

The potential risks and costs of an evacuation are numerous and can change depending on the location. The unpredictable and complex nature of humans, however, is a constant. Lack of preparation, inability or refusal to follow directions, road rage, and accidents can cause confusion on the roadways, leading to congestion and longer evacuation times. Accordingly, evacuation related deaths have been linked to accidents and health issues such as heat stroke. Some evacuation methods (e.g. segmented and contraflow) take these factors into account and attempt to limit the number of people on the road or increase the carrying capacity of the roads. The costs that are associated with evacuations must also be considered, with the most worrisome being the loss of lives. Hurricane Rita, after which many casualties were attributed to the evacuation, exemplifies the potential risks and costs of evacuation. Currently, our model does not take these issues into account, as it is a maximum flow model.

The two methods simulated beyond chaotic evacuation, segmented and contraflow, are based on real-world methods. The first, segmented evacuation, is used most notably in Florida, where the state is divided into several regions. In case of an emergency, the regions are given specific dates and times to evacuate to ease the burden on the roadways [6]. This is most relevant during large-scale evacuations. To reflect this, the road system in our Netlogo model, may be divided into sections via a slider. The second method, contraflow evacuation, involves converting major roads and those parallel to them into one-way streets, repurposing the other direction to increase traffic flow[7].

Description

Scope

Our two-dimensional Netlogo model simulates the evacuation of a population from an at-risk area or to an area of safety through a simulated road system. This enables us to compare the time taken for complete egress via several modes of evacuation. The road system is created using a combination of agents and links, representing nodes (or intersection) and roads respectively. Nodes have variables that allow them to track their distance from the source to aid in finding routes. Links, on the other hand, have a mile per hour (mph) variable that represents the speed limit along that road.

Currently, two real-world methods of structured evacuation have been implemented, as well as a chaotic evacuation. The first structured evacuation model, based on the method used in Florida, is segmented. Recall, this involves the designation of regions or sections which are evacuated separately based on risk and proximity to the threat. The second structured method is contraflow evacuation. This mode of egress requires the conversion of major roads and those that run parallel to them into one-way streets running in the direction of the desired exit. Within our program, the chaotic method of evacuation provides the foundation for both models. In chaotic evacuation, the entire area exits via the most efficient path as determined by a flood-fill algorithm. All modes of egress will be explained in greater detail in the *Methods* section below.

Setup and Scaling

To control the experiment, both the layout of the intersections and the speed limits of the roads were standardized. The nodes are distributed in a square grid with the distance between them established by the user before running the model. This allows for expansion or contraction of the area to be simulated for more versatile data collection and testing. Standardization of the speed limits made use of their location in the Netlogo world. This provided a distribution of values to demonstrate the functionality of the model.

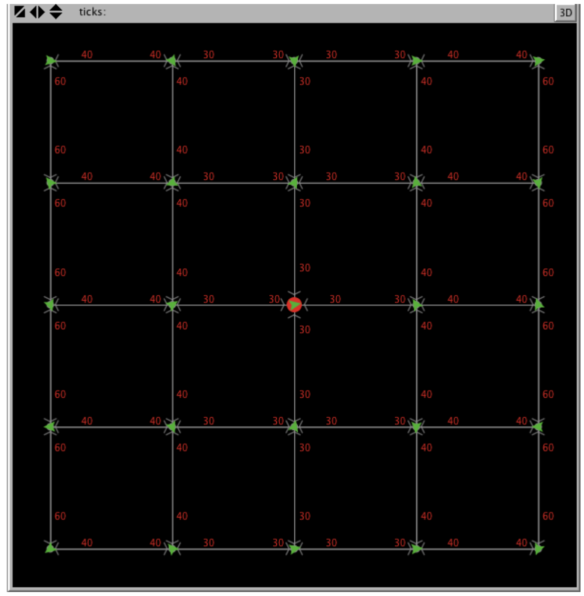

Figure 5: Screenshot of expanded grid layout

To analyze the results two assumption were made on the scale of the model. Each "step" or cycle the model ran through was decided to be 1/1000 of an hour. This means each car in one cycle traveled the speed limit of the link they were on divided by 1000. To further simplify this math and retain clean numbers, each patch was set as 1 square mile measuring one mile horizontally and vertically. From these numbers the amount of cycles taken to complete an evacuation by the last car to reach the goal, or the furthest point, can be converted into a time in minutes by dividing by 1000 and then dividing by 60.

Methods

As stated above, the simulation models several methods of evacuation. However, before any of these can be run, the program completes a flood-fill to determine the distance of every node from the source node which represents either a destination or danger. The flood-fill algorithm is a recursive function originally called by the source and then run by all linked nodes. First every node sets its distance from the source to an arbitrarily high value greater than any possible distance. This step is necessary for the algorithm to function. The source then sets its variable which records the distance from source to zero. Finally, it asks all connected nodes with a value for their distance from source greater than its own, to add their distance from itself to its distance from source and then run the same program. By having only those with a higher value run the program, it ensures that the lower values, indicating distances already calculated, are not overwritten. When these next nodes run the same program, they have their connected nodes add their distance from the asking nodes to that nodes distance source. With each successive node that runs the program, the values for their distances are filled in a flood outward from the source. Once there are no more nodes with values higher than the previous nodes, the program completes and the methods for evacuating can be run.

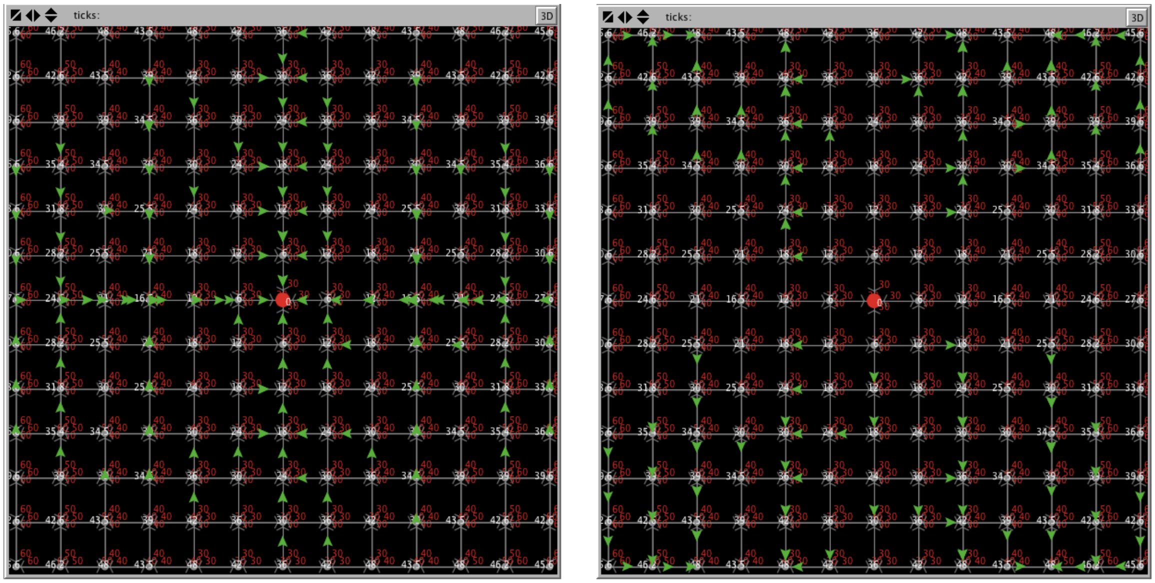

Figure 6: Screenshot of evacuate-to and evacuate-from in process

The two modes of structured evacuation, segmented and contraflow, both rely on the systems of chaotic evacuation, evacuate-to and evacuate-from. In these systems, the cars select their next target intersection, or goal, from those nodes connected to their current node. For evacuate-to, the node with the lowest value for distance from source is chosen, as choosing a lower value will eventually lead it to zero (0), the value held by the source. Conversely, in evacuate-from the node with the highest value is chosen. In segmented evacuation, the road system is divided into several segments as determined by a slider. Then the cars within each region, progressing from bottom to top and left to right, run either evacuate-to or evacuate-from. Once a region is fully evacuated,

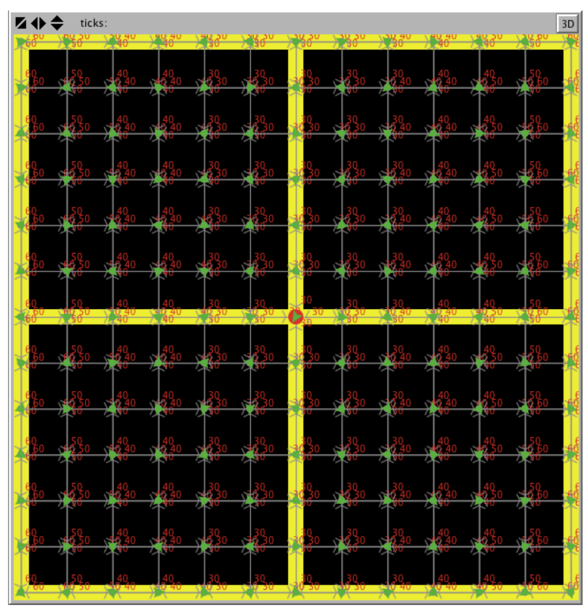

Figure 7: Screenshot of segment grid visualization

indicated by an absence of cars, the vehicles in the next segment run the same method and so forth and so on until no more need to be evacuated. For contraflow evacuation, the change is made before the flood-fill algorithm is run. The links with a heading approximately equal to the number representing one of the cardinal directions (e.g. 0 for north and 90 for east) die leaving only the link pointing in the opposite direction. Then the flood-fill and chaotic functions are run. It should be noted, that it is possible to perform both methods simultaneously, but for comparison they were only run separately.

Challenges

Throughout the development of the code, several challenges had to be addressed and the code's scope changed accordingly. The first and frequently recurring problem was the use of unfamiliar Netlogo functionalities, specifically links. Due to this, the code is not as fully developed as planned; self-teaching and communication with mentors to gain the necessary knowledge delayed the coding process. This delay prohibited the inclusion of more complicated variables and behaviors such as human actions, inhibiting the model's real-world applicability.

The most significant coding issue occurred during the development of the algorithm for contraflow evacuations. To establish one-way roads, directed links were used which only respond and interact in one direction. Once those links were implemented, however, the flood-fill program and chaotic evacuation methods stopped functioning. This was due to an incompatibility between directed and undirected links that required a light, but expansive, reworking of the entire code to ensure that all parts were able to function together.

Conclusions

Results

To gather consistent data, several governing decisions were made. First, the distance between intersections was set at 3, providing a large road system. Second, only four segments were used for segmented evacuation; up to 36 are possible. Finally, only one of the four potential contraflow evacuation methods were used as they produce nearly identical results. With these parameters, the chaotic evacuation and contraflow evacuation for evacuate-to are nearly identical in the time taken for complete egress, averaging 78.140 minutes and 78.540 minutes, respectively. Evacuate-to segmented evacuation was drastically greater at 244.488 minutes. This number is roughly proportional to the number of sections evacuated as compared to chaotic evacuation.

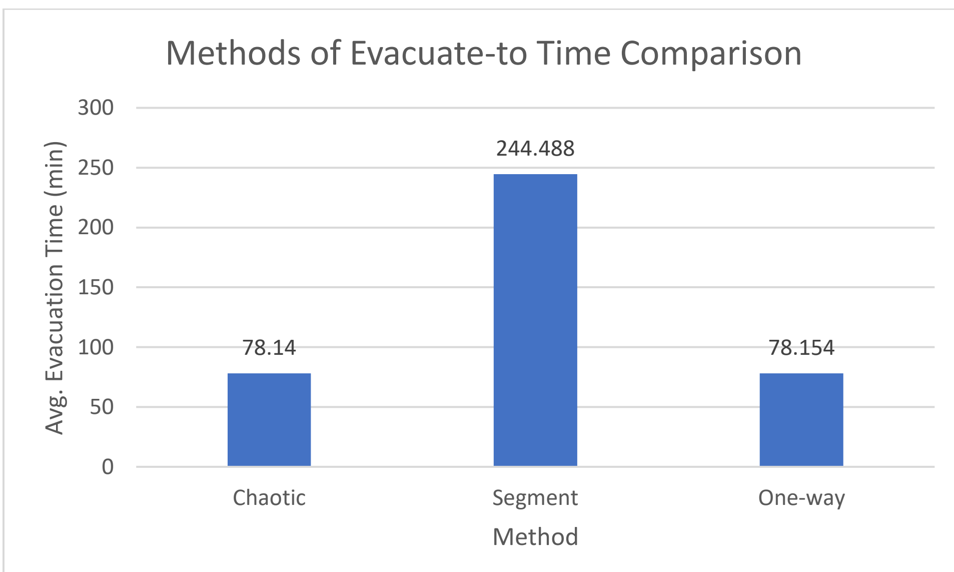

Figure 8: Methods of Evacuate-to Time Comparison Graph

The evacuate-from data includes only the chaotic and segmented modes which are similarly far apart. The former recorded 89.707 minutes and the latter 268.800

minutes. Contraflow was not modeled as it is a method of evacuating to a point outside the city.

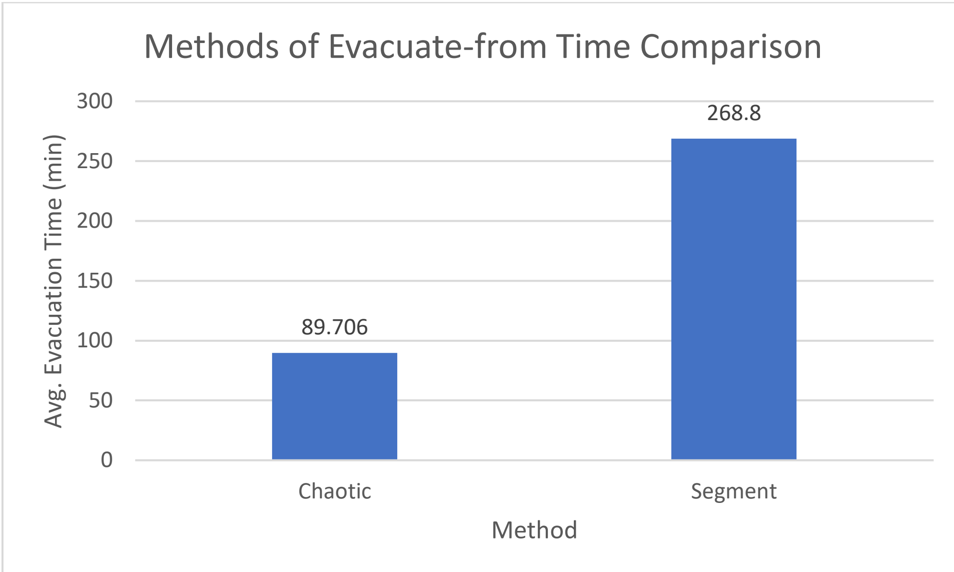

Figure 9: Methods of Evacuate-from Time Comparison Graph

Analysis and Future Plans

The data gathered from our model provides a basic understanding of the comparative times associated with each method. However, as the model does not account for human behavior and other complex variables the results it produces are not applicable to the real-world. This is further illustrated by the narrow range observed in the results. Due to the nature of the methods at this time, the results produced are computable and standard. To expand, as the cars simulated always choose the best routes and are unimpeded, the maximum time is roughly equivalent to the furthest agent. For the model to represent the real world, human behavior and its consequences, such as traffic and accidents, need to be implemented. These have the potential to drastically affect the time taken to evacuate in all cases. In the meantime, the model as it currently stands, could be utilized as a foundation for more complicated iterations.

References

[1] “Evacuation Planning in Texas: Before and After Hurricane Rita.” *Interim News*, 14 Feb. 2006, www.hro.house.state.tx.us/interim/int79-2.pdf.

[2] “Why Was Houston Not Evacuated before Hurricane Harvey?” BBC News, BBC, 29 Aug. 2017, www.bbc.com/news/world-us-canada-41085438

[3] “Hurricane Irma: Time Runs out for Florida Evacuation.” BBC News, BBC, 10 Sept. 2017, www.bbc.com/news/world-latin-america-41214622.

[4] Schneider, Avie. “Southern California Fire Forces New Evacuations, Threatens Coastal Communities.” NPR, NPR, 17 Dec. 2017, www.npr.org/sections/thetwo-way/2017/12/16/571380748/southern-california-fire-forces-new-evacuations-threatens-coastal-communities.

[5] Li, et al. “Optimization and Planning of Emergency Evacuation Routes Considering Traffic Control.” The Scientific World Journal, Hindawi, 29 May 2014, www.hindawi.com/journals/tswj/2014/164031/.

[6] “Emergency Information.” FloridaDisaster.org, Florida Division of Emergency Management, www.floridadisaster.org/info/.

[7] MCEMA. “Contraflow.” *Mobile County EMA > Evacuations Procedures > Contraflow*, 2018, www.mcema.net/Evacuations-Procedures/Contraflow.

[8] Wang, Jing-Hong. “Principal Aspects Regarding to the Emergency Evacuation of Large-Scale Crowds: A Brief Review of Literatures Until 2010.” Procedia Engineering, Elsevier, 16 May 2014, www.sciencedirect.com/science/article/pii/S1877705814004196.

[9] Griggs, Brandon. “Harvey's Devastating Impact by the Numbers.” *CNN*, Cable News Network, 1 Sept. 2017.

[10] Levin, Matt. “How Hurricane Rita Anxiety Led to the Worst Gridlock in Houston History.” *Houston Chronicle*, 22 Sept. 2015.

[11] Modesti, Kevin. “Assessing the Scope: Southern California Fires Add up to Disaster on a Jaw-Dropping Scale.” *Daily News*, Daily News, 9 Dec. 2017.

[12] Shaw, John R. “QuickFill: An Efficient Flood Fill Algorithm.” *QuickFill: An Efficient Flood Fill Algorithm. - CodeProject*.

[13] Wang, Amy B, et al. “As Harvey Submerges Houston, Local Officials Defend Their Calls Not to Evacuate.” *The Washington Post*, WP Company, 28 Aug. 2017.

Acknowledgements

Thank you to all those below…

Ms. Hope Cahill, a teacher at El Dorado Community School, provided regular brainstorming, organizational, and editing support throughout the entire project. She made sure that the team kept focus during our meetings and helped us keep on track. This project would not be completed without her.

Mr. Brian Smith was extremely helpful in the initial organization of the team and keeping it on track to finish. He made sure that we had the funds to register for the Supercomputing Challenge and participate in the Kickoff.

Mr. Stephen Guerin, who taught us how to use links and nodes in Netlogo, a system we were unfamiliar with, to the point of never having heard of it. And who helped us get the code off the ground by guiding us towards using a flood-fill algorithm to solve our pathfinding dilemma. He also pointed us in the direction of libraries we wouldn't have accessed otherwise.

Team Biographies

Rowan Cahill is a senior attending Santa Fe High School and has an interest in all things mathematical. He will be attending Colorado School of Mines to study Computer Science and Mechanical Engineering. This year, Rowan was primarily responsible for the development of the code and the methods used to simulate the evacuations. He also did extensive research and segments of the paper and final edit.

email: rcahill505@gmail.com

Ramona Park is a senior at Santa Fe High School who holds interests in graphic design and the production of creative content. She is aimed at pursuing a degree in computer science with an emphasis on communication and visual arts in order to implement better flow of information in the digital age. During this project, she played a central role in organizing our resources and contributed to both the interim and final reports.

email: yoharamona@gmail.com

Luke Shankin is a senior at Santa Fe High School who will begin attending New Mexico Tech in the fall of 2018 in order to pursue a degree related to Computer Science. Over the course of the Challenge, Luke played a role in leadership, conducted research related to historical evacuations, worked with the code, and contributed to both the interim and final reports.

email: lukeshankin@gmail.com

Appendix A: Data Tables

Chaotic Evacuation: To	
Trial	Evacuation Time (min)
1	77.88
2	77.88
3	77.88
4	77.88
5	77.88
6	77.88
7	77.76
8	77.88
9	80.6
10	77.88
Avg.	78.14

Chaotic Evacuation: From	
Trial	Evacuation Time (min)
1	89.76
2	89.7
3	89.76
4	89.64
5	89.76
6	89.76
7	89.58
8	89.64
9	89.7
10	89.76
Avg.	89.706

Segment Evacuation: To	
Trial	Evacuation Time (min)
1	242.28
2	242.34
3	242.28
4	249.6
5	246.72
6	242.28
7	242.16
8	246.78
9	248.16
10	242.28
Avg.	244.488

Segment Evacuation: From	
Trial	Evacuation Time (min)
1	262.38
2	269.04
3	273.6
4	268.98
5	273.36
6	266.1
7	273.36
8	267.6
9	266.1
10	267.48
Avg.	268.8

Contraflow Evacuation: To	
1	77.8
2	80.76
3	77.88
4	77.88
5	77.88
6	77.88
7	77.88
8	77.88
9	77.88
10	77.82
Avg.	78.154

Contraflow Evacuation: To (South)	
1	77.88
2	77.88
3	77.88
4	80.76
5	77.88
6	77.88
7	77.88
8	77.82
9	77.76
10	83.7
Avg.	78.732

Appendix B: Code

```
breed [node]
breed [cars]
breed [null]
directed-link-breed [roads road]
globals [#steps spacing x x2 y y2]
node-own [source timesource]
links-own [mph]
cars-own[originx originy goal speed counter distancetraveled timetraveled]

to setup
 ca
 set #steps 0
 ask patches
 [
  if ((pxcor mod int-freq) = 0 and (pycor mod int-freq) = 0)
  [
   sprout(1)
   [
    set breed node
    set shape "circle"
    set color grey
    set size .5
   ]
  ]
 ]
 ask node
 [
  set timesource 9999
  create-roads-to other node with [distance myself <= (int-freq)] ;finds an adjacent node and
makes a link.  Change to <= (int-freq * sqrt(2)) or equal to for diagonals
  hatch(1)
  [
   set breed cars
   set color green
   set size 1
   set goal node with [distance myself = 0]
   set originx pxcor
   set originy pycor
  ]
 ]
 ask roads
 [
```

```
    set mph 30 + round(([pxcor] of end1 * [pxcor] of end2) / 100) * 10 ; standardizing mph of
roads
    set label precision mph 2
    set label-color red
  ]
  repeat num-source ;disabled for data gathering
  [
    ask node with [pxcor = 0 and pycor = 0] ;ask one-of node with [color != red]
    [
      set source true
      set color red
      set size 1
    ]
  ]

  run[segment-setup]
end

to flood
  ask node
  [
    set timesource 9999
  ]
  ask node with [source = true]
  [
    set timesource 0
    flood-fill
  ]
  ask node
  [
    set label precision timesource 2
  ]
end

to from-
 ask cars
  [
    if ([breed] of goal != node or distance goal <= .06)
    [
      let ngoal choose-goal-more
      if (ngoal != nobody)
      [
        let temps 0
        ask goal
        [
```

```
        set temps ([mph] of link-with ngoal) / 1000
       ]
       set speed temps
     ]
     set goal ngoal
   ]
   ifelse (goal != nobody)
   [
     face goal
     forward speed
     set counter counter + 1
   ]
   [
     if counter > #steps
     [
      set #steps counter
     ]
     set breed null
     set size .05
   ]
  ]
  wait .001
end

to to-
  ask cars
  [
   if ([breed] of goal != node or distance goal <= .06)
   [
     let ngoal choose-goal-less
     if (ngoal != nobody)
     [
       let temps 0
       ask goal
       [
        set temps ([mph] of link-with ngoal) / 1000
       ]
       set speed temps
     ]
     set goal ngoal
   ]
   ifelse (goal != nobody)
   [
     face goal
     forward speed
```

```
      set counter counter + 1
    ]
    [
      if counter > #steps
      [
       set #steps counter
      ]
      set breed null
      set size .05
    ]
  ]
  wait .001
end

to to-from
end

;----------------------------------------------------------------------------------------------------

to north-orient
  ask node
  [
    let origin self
    let temp 0
    ask my-out-links with [abs(link-heading - 180) <= 10]
    [
      die
    ]
  ]
  ask node with [source = true]
  [
    set source false
    set color grey
    set size .5
  ]
  ask node with [pxcor = 0 and pycor = [ycor] of max-one-of node [ycor]]
  [
    set source true
    set color red
    set size 1
  ]
end

to south-orient
  ask node
```

```
  [
    let origin self
    let temp 0
    ask my-out-links with [abs(link-heading - 0) <= 10]
    [
      die
    ]
  ]
  ask node with [source = true]
  [
    set source false
    set color grey
    set size .5
  ]
  ask node with [pxcor = 0 and pycor = [ycor] of min-one-of node [ycor]]
  [
    set source true
    set color red
    set size 1
  ]
end

to west-orient
  ask node
  [
    let origin self
    let temp 0
    ask my-out-links with [abs(link-heading - 90) <= 10]
    [
      die
    ]
  ]
  ask node with [source = true]
  [
    set source false
    set color grey
    set size .5
  ]
  ask node with [pycor = 0 and pxcor = [xcor] of min-one-of node [xcor]]
  [
    set source true
    set color red
    set size 1
  ]
end
```

```
to east-orient
  ask node
  [
    let origin self
    let temp 0
    ask my-out-links with [abs(link-heading - 270) <= 10]
    [
      die
    ]
  ]
  ask node with [source = true]
  [
    set source false
    set color grey
    set size .5
  ]
  ask node with [pycor = 0 and pxcor = [xcor] of max-one-of node [xcor]]
  [
    set source true
    set color red
    set size 1
  ]
end

to segment-setup
  if (segments != 0)
  [
    set spacing (max-pxcor / (segments / 2))
    if (borders = true)
    [
      ask patches with [(pxcor mod spacing) = 0 or (pycor mod spacing) = 0]
      [
        set pcolor yellow
      ]
    ]
  ]
end

to to-segment
  set x min-pxcor
  set x2 (x + spacing)
  set y min-pycor
  set y2 (y + spacing)
```

```
 while [x != max-pxcor]
 [
  while [y != max-pycor]
  [
   while [count cars with [originx >= x and originx <= x2 and originy >= y and originy <= y2]
!= 0]
   [

    ask cars with [originx >= x and originx <= x2 and originy >= y and originy <= y2]
    [
     if ([breed] of goal != node or distance goal <= .06)
     [
      let ngoal choose-goal-less
      if (ngoal != nobody)
      [
       let temps 0
       ask goal
       [
        set temps ([mph] of link-with ngoal) / 1000
       ]
       set speed temps
      ]
      set goal ngoal
     ]
     ifelse (goal != nobody)
     [
      face goal
      forward speed
      set counter counter + 1
     ]
     [
      if counter > #steps
      [
       set #steps counter
      ]
      set breed null
      set size .05
     ]
    ]
    wait .001

   ]
   ask cars
   [
    set counter #steps
```

```
      ]
      set y y2
      set y2 (y + spacing)
    ]
    ask cars
    [
      set counter #steps
    ]
    set y min-pycor
    set y2 (y + spacing)
    set x x2
    set x2 (x + spacing)
  ]
end

to from-segment
  set x min-pxcor
  set x2 (x + spacing)
  set y min-pycor
  set y2 (y + spacing)

  while [x != max-pxcor]
  [
    while [y != max-pycor]
    [
      while [count cars with [originx >= x and originx <= x2 and originy >= y and originy <= y2]
!= 0]
      [

        ask cars with [originx >= x and originx <= x2 and originy >= y and originy <= y2]
        [
          if ([breed] of goal != node or distance goal <= .06)
          [
            let ngoal choose-goal-more
            if (ngoal != nobody)
            [
              let temps 0
              ask goal
              [
                set temps ([mph] of link-with ngoal) / 1000
              ]
              set speed temps
            ]
            set goal ngoal
          ]
```

```
        ifelse (goal != nobody)
        [
          face goal
          forward speed
          set counter counter + 1
        ]
        [
          if counter > #steps
          [
            set #steps counter
          ]
          set breed null
          set size .05
        ]
      ]
      wait .001

    ]
    set y y2
    set y2 (y + spacing)
  ]
  set y min-pycor
  set y2 (y + spacing)
  set x x2
  set x2 (x + spacing)
 ]
end

;----------------------------------------------------------------------------------------------------

to-report choose-goal-less
 let temp nobody
 if goal != nobody
 [
  ask goal
  [
   let possible out-link-neighbors with [timesource < [timesource] of myself] ;for consideration
of mph as well: timesource * [mph] of link-with myself (NEEDLESS IF TIME ALREADY
CONSIDERED IN FLOODFILL)
   set temp min-one-of possible [timesource]
  ]
 ]
 report temp
end
```

```
to-report choose-goal-more
  let temp nobody
  if goal != nobody
  [
    ask goal
    [
      let possible out-link-neighbors with [timesource >= [timesource] of myself]
      set temp max-one-of possible [timesource]
    ]
  ]
  report temp
end

to flood-fill
  ask in-link-neighbors [
    let possibletime ((distance myself / [mph] of link-with myself) * 60) + [timesource] of myself
;for consideration of mph: (distance myself / [mph] of link-with myself)
    if possibletime < timesource
    [set timesource possibletime
     flood-fill
    ]
  ]
end
```