Paper 03:
Modeling urban evacuation process

Author: **Jiwon Baik**, University of California Santa Barbara, jiwonbaik@ucsb.edu

Introduction

Evacuation is an important process to save people from the danger of fire and other hazards. With the limited resources, confined road network, and network's capacity, it's important to consider the efficiency of the evacuation process in urban spaces. In addition, potential hardships in the process such as accidents and congestion should be acknowledged beforehand, so that they can be handled in a real-time evacuation situation. By simulating the urban evacuation process, experiments that cannot be conducted in the real world could be done. For example, an intersection can be closed according to an accident, and a private road could be opened during the evacuation time to see how it will impact the time required to make the whole region empty and let the whole population evacuate.

There's a general assumption related to the evacuation simulation. Since evacuees are familiar with the surrounding common environment, they will always choose the optimal evacuation routes, which leads to the optimal evacuation process of the whole region. However, this assumption doesn't turn out to be always true. Individual drivers' primary goal is to evacuate themselves, not to help the whole people in the region to evacuate as soon as possible (in other words, to minimize the region's clearance time). When all people take the same shortest path to reach a destination, it might cause congestion. When people enface congestion, they might choose another route that is not the shortest path. As an individual driver can only know about the road they are on, such a detour can cause a delay in the region's clearance time. Moreover, recent studies have pointed out that the emotional status of individuals and cognitive behavior can affect evacuees' movement (Brachman, 2014; Epstein et al. 2011; Fang and Aguirre, 2016a; 2016b). For example, not all people choose to evacuate. Some people choose to stay home because they don't think the fire wouldn't arrive at their home. When fire or air pollutants are visible, evacuees might become more frustrated, and it will impact their driving behavior. It can consecutively impact the whole region's clearance time. Such concerns provoked a need for a more complex simulation of the evacuation process incorporating individual driving behavior.

Common existing approaches to address fire evacuation can be divided into two parts, field study, and simulation. During or after the emergency situation, surveys asking about the evacuation experience can be sent out (Church and Sexton, 2002). Usually, surveys ask about demographic attributes and spatial information. Spatial information includes the origin and the destination, and the routes they took during the evacuation process. The responses become invaluable ground-truth for the second approach, simulation. Depending on the size of the study region, the simulation could be in a macro or micro setting (Langford, 2010). If the study region is relatively small as including only thousands of houses, each vehicle moving along the road network can be simulated. Each vehicle can be represented as a point, or as a polygon to be more detailed. Usually, commercial transportation microsimulation software as VISSIM and Paramics represent each vehicle and each road segment as a polygon sothat every detail of driving behavior can be modeled, for example, velocity changes at curves or near the other vehicles. As the study region gets larger as a neighborhood or a city, including more population, it becomes computationally not feasible to represent each vehicle. Thus, flow in each road segment can be modeled. For example, an agent-based transportation modeling software, MATSIM, simulates the flow of each road segment. MATSIM avoids tracking individual behavior but therefore enables tracking macro-scale movement throughout the whole city. During each time interval, the software models the average velocity of each road segment defined by the origin and destination nodes.

The micro-scale simulation software, VISSIM, and Paramics are commercial software developed in the 1990s, but the software hasn't been updated since the companies have been closed. It's hard to get access to the software and especially gives a hard time for users to build the road network in the software as it needs the polygon format of the road network. It's computationally heavy relying on the polygon representation of vehicles and roads, sothat only a small region can be simulated. However, the meticulous representation might not be accurate and hasn't been proved enough whether it helps reproduce the real-world operation. The macro-scale simulation software, MATSIM is open-source software and keeps being updated (the latest update is April 2021) and can cover a region even larger than a complex city. However, the macro scale simulation does not incorporate individual driving behavior. All three software mentioned above are not particularly targeting the evacuation process but general transportation simulation which doesn't allow any variation in the emotional status or cognitive behavior of drivers. Therefore, this paper suggests an agent-based simulation of the evacuation process with the point, not polygon, representation of each vehicle and line-node representation of road network, incorporating emotional and cognitive behavior or drivers according to the surrounding environment.

Background

There have been two different modeling approaches in the evacuation simulation. One is mathematical models and the other is agent-based simulation. The mathematical models are from the field of operations research. Langford (2010) added the time dimension to the two-dimensional road network. In Langford's model, the total transportation cost, and the total clearance time of the region are minimized. In the space-time network, the number of vehicles moving through each road segment at each timestep could be tracked. But it's not tracking the individual vehicle. So the Langford's model can be classified in macro-scale simulation as MATSIM software discussed in the introduction. Brachman proposed two different evacuation models in 2012 and 2014. Both models are based on the typical network flow models and minimal cost models in the operations research. So both models don't track timely movement but find the best network flow in terms of the different cost objectives. Brachman (2012) proposed a mathematical model to simulate the evacuation process in the context of minimizing the total cost for evacuees traveling along with the road network, plus the total traffic congestion cost. The model can incorporate a situation where a driver detours, giving up the shortest path when they meet congestion in the shortest path. However, it still relies on the optimization context where individual drivers can make the optimal decision considering the whole road network status, which is unrealistic. Also, the detour decision relies on the pre-defined congestion score of each road segment, regardless of the autonomous movement of vehicles and the emotional status of the individual driver. From this paper, the Rayleigh probability model was proposed to model how much percentage of the total population of the region will depart at each time step. This approach was compared with the survey result from the Jesusita fire case and proved to reflect the real-world phenomena well. Rayleigh's probability model takes one parameter to define the probability function. Figure 1 shows how the probability function changes according to the different parameter values. The user gives information to define three parameters indicating the fire hazards of the whole region regarding the fire distance, weather, evacuation alarm. f_{bar} is the average value of the three parameters, a higher value means more threat. It defines the Rayleigh probability function. It should be calibrated as the study region changes.

f_1 *=3 if the fire has burned homes*
2 if the fire poses a significant threat to homes
1 if the fire poses a minimal threat to homes

f_2 *=3 if wind speed and air temperature are extreme*
2 if wind speed and air temperature are above average
1 if wind speed and air temperature are at or below average

f_3 *=3 if all residents have been ordered to evacuate*
2 if some residents have been ordered to evacuate

1 no evacuation order issued

$f_{bar} = \frac{(f_1+f_2+f_3)}{3}$

Rayleigh Probability Model: $P_{evac} = exp(-T^{f_{bar}}/50)$

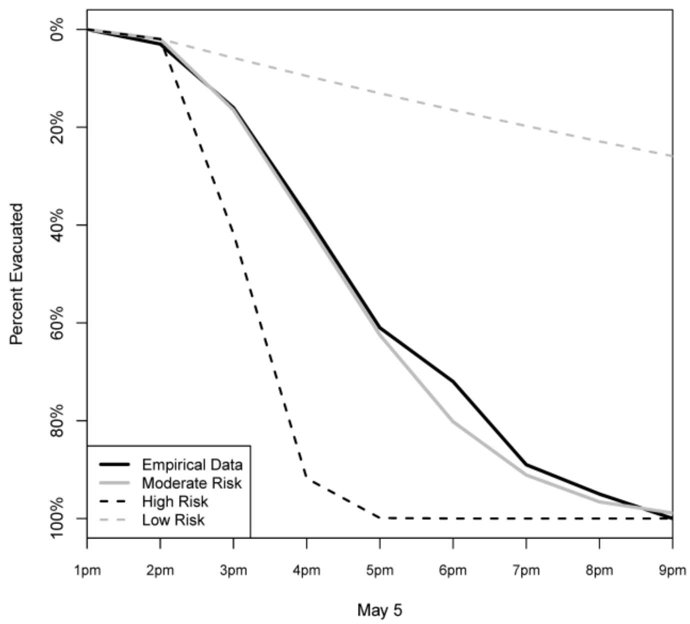

Figure 1. Rayleigh Probability Model with different fbar values (source: Brachman, 2012)

Brachman (2014) proposed another evacuation process model which minimizes the cost in terms of the cars enfacing the fire hazards along with the road network. The different levels of fire hazards were given as static ellipses, and the model tries to find the network flows which minimizes the flow going through the dangerous region. Again, individual emotional status or timely driving behavior cannot be tracked in this model.

Among the three mathematical models related to the evacuation process, Langford (2010) gives insight the most as it's a Spatio-temporal model at least tracking the macro flow. This paper will track the individual vehicle on the micro-scale. Especially, the way road network is defined will follow Langford's work. Also, the concerns about the cognitive behavior related to the congestion and fire hazards covered by Brachman (2012, 2014) will be incorporated into this paper in a timely and autonomous manner. Each driver will autonomously determine how to react when they encounter congestion or dangerous roads. And the movement will be simulated by each timestep. Now in a total flow as Brachman did.

There are not so many agent-based modeling works related to the urban evacuation process especially exploiting a real road network. There is some commercial or open-source transportation simulation software discussed in the introduction section, but not so much paperwork has been published to validate the reliability of the software due to the lack of ground truth data. Especially, the software doesn't consider the emotional status of individual drivers. Chen and Zhan (2003, 2016) conducted the agent-based simulation with the real road network in the evacuation process with Paramics. The paper gives good hints on what to consider in evacuation process simulation such as the aggressiveness of the drivers, car-following behavior rules, and different driving modes. However, by relying on the pre-built software, no behind-the-scenes analysis has been done. There's no mathematical formula to specify the relationships. The probabilistic approach is applied to define the aggressiveness of each driver at the beginning of the simulation, but the figure is fixed during the whole simulation time. So no cognitive changes can be captured by the approach. Moreover, the calibration or validation has not been done in the works but only simulation results were presented. Fang and Aguirre (2016a; 2016b) simulated egress procedure with agent-based modeling. Egress is an evacuation process in the building, not in the road network. Though the context is different, the papers give a good insight to model the cognitive behavior of individual agents. Each agent has a scalar vector indicating the emotional status in different categories, which changes as the surrounding environment changes as the agent moves around the building. And the scalar values control the behavior of the agents, which reflects the emotional and cognitive behavior into the model. This paper takes the same approach assigning the scalar vector to each individual driver which affects the driving behavior.

Conceptual Framework

1. Model Input

The model needs two shapefiles and one road network as an input. The two shapefiles are to define the f_1, and f_3 values for the Rayleigh probability model introduced in the Background section. The fire perimeter shapefile will be used to determine if any house was burnt when the simulation starts. When any house was burnt, f_1 becomes 3 as defined, if no house was burnt but the closest house is within 100m from the fire perimeter, f_1 becomes 2. If not, f_1 becomes 1 with the least danger. The 100m threshold can be changed through calibration. The people living in the Evacuation Zone will receive the Evacuation alarm. If all houses are within the Evacuation Zone shapefile, $f_3$value will be 3 as defined, if more than 1 house is within the Evacuation Zone shapefile, $f_3$value will be 2 as defined. If there's no house within the EZ shapefile, $f_3$will be 1. In this paper, $f_2$value will be fixed to 2, in order to reduce the number of parameters to be provided according to the law of parsimony.

The road network should be provided or a bounding box of the ROI should be provided. The model includes the pre-process of the road network dataset from the open data sources as OSM and Microsoft Building footprints to automatically build a road network of the region, and detect the source nodes where evacuees will depart. The road network consists of node-set and arc set. Each node has two attributes in addition to ID, which are supply and sink_node. When the supply attribute value is greater than zero, it means the node is the house node, where vehicles will depart. The value means the number of vehicles that will depart from the house node. The sink_node is a binary variable indicating whether the node is a sink node that is connecting the other road segments to the highway. When people arrive at a sink node, the evacuees can move to a distant place that is safe from the fire through the highway. The sinknode becomes the destination of the whole population. There can be multiple sinknodes in one road network. The road network will be processed from the raw data provided by OpenStreetMap and house nodes will be extracted from the building footprints provided by the Santa Barbara flood district.

Each vehicle can choose which sinknode will become their destination. By default, the closest sinknode will be chosen by the Breadth-First Search (BFS) algorithm. Each arc in the Arc set will be defined by the from_node and to_node IDs. Each arc has a capacity value, indicating the number of vehicles it can handle in each time step. One arc represents a road segment in the real world, and they are 80 meters long on average. It is based on the assumption that when vehicles are moving along the residential area to get to the highway, they are going to move in 8 m/s velocity on average. So in 10 seconds, they are going to move 80 meters. As the road network is processed in this way, to reduce the computational burden, the BFS algorithm is used to find the closest path, rather than the Dijkstra algorithm. Usually, one road segment can hold 5 vehicles in each time step, but the capacity reduces when there's a stop sign or traffic signal. At intersections, the more roads are connected to the intersection, the less capacity of the roads. These road network settings are from Langford (2010) and Church and Saxton (2002). In the implementation, the road network is defined as a Graph. And the length of an arc and an entity's velocity on the road network defines the number of timesteps the entity should stay on the road network.

Nodes {ID, supply, sinknode, depart_time, hold}
Arcs {from_node, to_node, capacity, length}

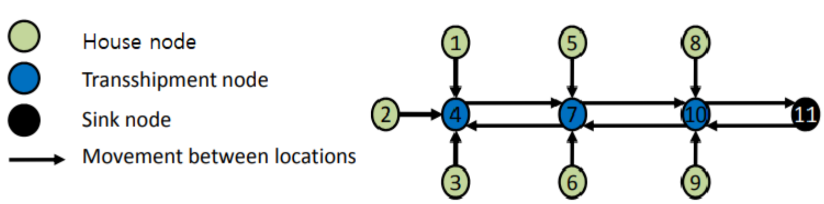

Figure 2. Simple Road network (source: Langford, 2010)

2. Entities

Static info: {ID, houseNodeID, depart_time}
Variable: { emotional_status, path, hold_count,
velocity, position_node, next_node, time_left}

The entities involved in this system are individual vehicles starting out from each house in the study region. The common goal of evacuating drivers is to move to the destination along with the road network. Their goal is not to leave the whole region empty as soon as possible. Their goal is to minimize the evacuation time of their (each) own. In one term they are competing to get to the same destination, but in another term, they are cooperating to move peacefully through the road network. Evacuee will have three static information and seven changing variables to track&control the movement. Each vehicle has a scalar vector consisting of three values, {Emotional status value, Hold count}. The Emotional status value indicates how aggressive or frustrated the evacuee is feeling. As the value increases, it is more likely for the driver to make a detour. When making a detour, at the closest intersection, another node will be chosen as the next_node, which is not in the current shortest path. Then, from the new next_node, another shortest path to the closest sink node will be found by the BFS algorithm. The path each vehicle will take is recorded and tracked by the Path variable assigned to each vehicle. It is firstly defined by the BFS algorithm when each evacuee is activated. The Hold count value will increment by 1, as 1 second goes on without entering into the next road segment due to the other vehicles firstly entered to the road.

Another set of variables, {velocity, position_node, next_node, time_left} is to track which arc the Evacuee is at, and how much time is left to finish the arc trip. Those are changed with velocity when it is entering into a new arc. Velocity's default value is 10, indicating 10m/s. The speed limit of the residential area is 25 mph, which is 11.18 m/s. So 10 is set as a default value. The velocity will change according to the number of vehicles already on the road arc the evacuee wants to enter. Position_node basically means the starting node of the arc the evacuee is on, and the next_node is the end node of the arc. The time_left is determined by the length of the arc_length divided by the velocity of the evacuee at the moment when it's entering. And the time_left decreases as time goes by until it becomes zero.

The interaction rules each vehicle follows are as follows. The vehicles cannot enter into a road segment when the capacity of the road is fulfilled. So as a vehicle moves across the road network, it should consider the road capacity and the number of other vehicles on the road. Importantly, when a vehicle witnesses severe congestion where 100% of the road capacity is fulfilled, it should hold still at the node it is staying and with the probabilistic decision making, could decide to take a longer route than the current shortest path at each time step of 1-second interval. The greater the Emotional status value, the more likely the driver is to take a detour. When a vehicle is on the road which is congested (more cars than 80% of the road capacity), the emotional status of the driver will increase, which means he or she is more frustrated, and the Velocity will decrease until 8m/s. When a vehicle is on the road which is not congested at all (fewer cars than 50% of the road capacity), the emotional status of the driver will decrease, and the Velocity will increment until 12m/s.

Model Input: - Shapefiles for Rayleigh Probability Model: Fire Perimeter, Evacuation Zone - Road network : Nodes{ID, supply, sinknode, depart_time, hold} Arcs{from_node, to_node, capacity, length}
Entities: - Velocity: default 10m/s (close to 25 mph which is a speed limit of the residential area) - Emotional status scalar value: 1 - Path: default found by Breadth-First Search(BFS) algorithm - Hold count: default 0
Entities and Environment Interaction Rules: - First come, First served in each road segment - A vehicle can enter into the road segment only when the number of vehicles in the road segment is less than the road capacity - Otherwise, the vehicle should stay stop at the current node.   - Then, the Hold count value will increment by 1

Entities Interaction Rules:
- When there are more vehicles than 80% of the capacity of the road (congested)
    - the Emotional status value will increase
    - the velocity decreases to 8m/s
- When there are fewer vehicles than 50% of the capacity of the road (congested)
    - the Emotional status value will decrease
    - the velocity increases to 12m/s
- As the Hold count value increases,
    - the Emotional status value will increase by random
- When the Hold count is greater than 0,
    - Make a probability choice regarding Emotional status whether to make a detour or stick to the shortest path

Table 1. Model Interaction Rules Summary

3. Pseudocode of the Step functions

This section is to summarise what will happen in each time step, with a 1-second interval. A more detailed explanation can be given about how the interaction rules are being controlled in the model mathematically. In the perspective of the Model, it should check how many vehicles are on each road. Then it should check how many vehicles are arrived at the sink nodes, and make each evacuee do what they should do in each time step. Also, it should activate the evacuees who should depart at each time step. The process is described in Table 2 in the format of Pseudocode.

Model's step function

T = current time step
 	
For arc in arcs:
   Check the number of vehicles already on the arc
 	
For Evacuee in Evacuees:
   Check arrivals
   Evacuee's step function()
 	
For Evacuee in T.depart:
   Evacuee.activate()

Table 2. Pseudocode of the Model step function

Table 3 shows the Pseudocode of the Agent step function. At each second, the evacuee should do the following. If should check whether it's arrived firstly. If not, they should check if their path on the current arc is done or not. When it's still on the arc, the only thing it needs to do is move with the current velocity which was determined at the time it entered the road segment. So the time_left decreases by 1. When it is done with the current arc so it has to enter into the new arc, there are more things to be done. The position_node and next_node information should be changed according to the path information. According to the current hold count and emotional status, whether or not to find the new path by detour will be determined. As described, when the evacuee ensures that it's going to make a detour, at the next intersection it's going to choose any other node connected to the intersection, which is not in the current path. Then, the BFS algorithm will be used to find the closest path to the sink node from the newly chosen node. So the path information can be updated.

When the arc an Evacuee's trying to enter is congested, the velocity of the evacuee will be decreased by 1. But it shouldn't go under the 8m/s. When the arc is fully capacitated, the Evacuee should wait until there's room for it to enter. So only hold count and emotional status values will be incremented but the time_left will not be updated. The time_left value will be remaining zero, so it could go through a detour decision-making process with the increased hold count and emotional status values. When the arc is spacious, even without congestion, it could simply enter into the road with a velocity of at least 10m/s, the default velocity (25 mph). And the evacuee's allowed to speed up until 22m/s (50 mph) theoretically.

```
Evacuee's step function

Check arrival
   If path[-1] == Evacuee.next_node:
      Evacuee.arrive!

Check time left
   If time_left == 0:
      Evacuee.position_node = next_node
      Evacuee.next_node = next node in path
      Make_detour_or_not(Evacuee.holdcount, Evacuee.emotional_status)

      Check the next arc's status
         If arc.capacity * 0.8 < the number of vehicles on the arc <1: # congested
            Evacuee.velocity = max (8, Evacuee.velocity-1)
            Evacuee.holdcount = 0
            Evacuee.time_left =  next_arc.length / Evacuee.velocity

         Else if arc.capacity < the number of vehicles on the arc:    # should wait
             Evacuee.holdcount += 1
             Evacuee.emotional_status += 1

         Else:
            Evacuee.velocity = min(22, max(10, Evacuee.velocity+1))
            Evacuee.holdcount = 0
            Evacuee.time_left =  next_arc.length / Evacuee.velocity

   Else:
      Evacuee.time_left = Evacuee.time_left -1 # just move
```

Table 3. Pseudocode of the Agent step function

Implementation and Case Study

The ABM model for the urban evacuation process is built with Python library, MESA. And it's built on the jupyter notebook environment. Not all of the MESA functionalities have enough documentation or open-source education materials, so many parts had to be done manually. For example, the scheduler part of ABM is very important and also sophisticated. MESA currently has enough documentation of a Random scheduler or a Simple scheduler. The former activates each agent at a random time step, and the latter activates each agent following the order it is added to the model. Either is not appropriate for the urban evacuation model, because each evacuee agent has a predefined departure time found by the Rayleigh probability model introduced in the previous section. StagedActivation scheduler is an appropriate type of scheduler to be used, but there's no accessible application example or documentation to be referred to. Therefore, the model's time dimension had to be hand-coded whole part, which made the project challenging the most. (The ABM model is yet to be fully developed so the full implementation is yet to happen.)

The study region selected for the implementation is a mission canyon in Santa Barbara County, California. The region is defined by the North and South latitudes, and West and East longitude values (north = 34.4649585, south = 34.4450036, east = -119.7090010, west = -119.7248927). The road network inside the bounding box is downloaded from the OSM network and processed with python code to ensure it's in the forward-start matrix format. Also, the required node and arc set was created. The Santa Barbara Flood District provided the building footprints in Santa Barbara County sothat the source nodes were created from the same apartments. The model can handle any bounding box. When the bounding box region is out of Santa Barbara County, the building footprints will be found by the Microsoft Building footprint dataset. So the whole data processing part is from the open-source data. And it increases the user accessibility because the user doesn't need to process the road network data by themselves, but simply gives a bounding box input. The road segments are divided by 80-meter length intervals, and the new transshipment nodes will be added. The two-way roads will be duplicated in different directions. The intersection nodes will be separated into the in-node, and out-node. And there will be a new arc connecting the in-node and out-node. The driveway arcs will be created between the source nodes and the closest transshipment nodes along with the road network. In addition to that, stop signs and traffic signals are automatically detected from the google street view images of intersections. When there's a stop sign or a traffic signal, the capacity of the intersection will be systematically reduced related to the number of roads connected to the intersection. This process highly enhances the applicability of the model.

The dataset for calibration and validation is a result of the Microsimulation using a commercial transportation simulation software, Paramics. Table4 includes the result from the Church and Sexton (2002). The clearance time of the region can be different based on the number of cars activated. When the assumption is that from each house only one car departs, the optimal clearance time of the Mission Canyon region should be 13 minutes and 41 seconds. If the assumption is that from each house two cars depart, the optimal clearance time of the Mission Canyon region should be 32 minutes and 45 seconds. By the fact that it is a case of optimal operation, where all cars can see through the road status across the whole road network, the ABM result should be much longer than the microsimulation result. The ABM reflects the real-world operation where each evacuee can perceive the road status where they are at. In the implementation of Mission Canyon Region, both cases where one car or two cars depart from each house. And the result will be compared with Table 4. Of course, in the real world, more cars or no cars might depart from some of the houses, however, reality should be simplified in the model.

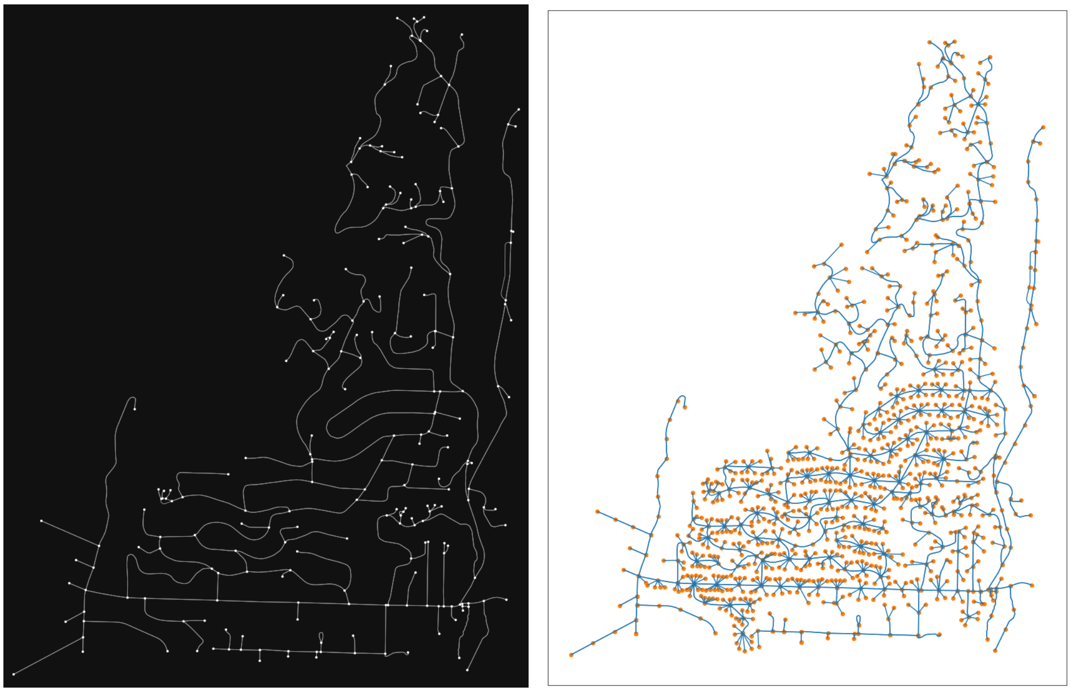

Figure 3. (left) original OSM data, (right) process road network with house nodes

		Vital Report	
		Church and Sexton (2002)	
		1car/house	**2car/house**
total cars		763	1526
% of cars	**50%**	8:23	15:43
	75%	12:04	24:16
	90%	15:28	30:25
	95%	16:44	32:40
	100%	18:49	34:58
# of cars	**200**	4:57	4:43
	400	9:14	8:47
	600	13:41	12:59
	800		16:55
	1000		21:54
	1200		26:53
	1400		32:45

Table 4. Microsimulation, Paramics, the result of the Mission Canyon region

In addition to this, there are 61 actual evacuation routes collected by the Jesusita Fire Survey. The Jesusita Fire Survey is an online survey of people in Santa Barbara, CA who were affected by the 2009 Jesusita wildfire. (Brachman, 2012) During the survey, respondents were presented with an interactive map interface and asked to click each road segment that they traveled on when evacuating from the fire. This map interface was followed by a text box in which evacuees were asked to write a turn-by-turn description of their evacuation route. In Mission Canyon Region, there are 141 routes including every survey response where the respondent clicked at least one segment on the interactive map. However, after validating the routes, comparing the route with the evacuee's starting road segment and end road segment, there were 61 routes left, validated. Unfortunately, the routes are not connected to the time information. So the departure time and arrival time of each route are not tracked. Therefore, the routes will be translated into a series of nodes. And it will be compared to the shortest path found by the BFS algorithm. Only 6 out of 61 routes had the same path as the shortest path. Therefore, the path will be used to calibrate the parameter controlling the probability of evacuee making the detour.

Conclusion and Future Work

This paper aimed to investigate how Agent-Based Modeling can be applied to the topic of urban evacuation planning. Agent-Based Modeling can be a breakthrough of evacuation planning research because it can track individual vehicle movements, without the surreal assumption that individual evacuees have the same goal as the whole community, and make the optimal decision based on the knowledge of the whole road network status. Moreover, it could be done with open-source datasets and software without paying a lot of effort and cost on commercial simulation software. Also, ABM can incorporate the emotional status and unreasonable decisions of the evacuee which is not applicable with general transportation simulation software. There have been many previous studies that worked with commercial software including mathematical solvers, or transportation simulators. But those were not able to capture individual movement, irrational wayfinding behaviors affected by emotional states, and often based on a wrong assumption overestimating the ability of individual drivers. Therefore, this paper has a significant and unique contribution in the field of urban evacuation planning study.

However, this work has a limitation in the conceptual part and also the implementation part. There are other factors than road status that can influence the emotional status of the drivers. For example, the distance to the fire and whether or not they can see the fire can impact their emotional status. This part should be incorporated to improve the quality of the evacuation process. To validate the component, empirical research on comparing the driving behavior of evacuees when they witness or do not witness the fire danger should be preceded. In the implementation part, there are not much empirical data to calibrate and validate the model. As the evacuation process is urgent, it's difficult to have the primary data from the evacuees, and of course, an experiment is impossible. The relationship between the emotional status and detour decision has not been discussed deep, so the probability function used in this paper is based on a simple guess which should be calibrated further. Additionally, as the time interval was set as one second, the model became very heavy to be handled with the hand-coding without a built-up scheduler. Time aggregation would be desirable if it's feasible. Whether or not time aggregation is reasonable should be determined by a sensitivity analysis comparing the results with different time intervals.

References

Brachman, M. L., & Dragicevic, S. (2014). A spatially explicit network science model for emergency evacuations in an urban context. Computers, Environment and Urban Systems, 44, 15-26.

Brachman, M. L. (2012). Modeling Evacuation Vulnerability. University of California, Santa Barbara.

Chen, X., & Zhan, F. B. (2014). Agent-based modeling and simulation of urban evacuation: relative effectiveness of simultaneous and staged evacuation strategies. In Agent-based modeling and simulation (pp. 78-96). Palgrave Macmillan, London.

Chen, X., & Zhan, F. B. (2003). Agent-based simulation of evacuation strategies under different road network structures. University Consortium of Geographic Information Science.

Church, R. L., & Sexton, R. M. (2002). Modeling small area evacuation: Can existing transportation infrastructure impede public safety?.

Eppstein, D. (1998). Finding the k shortest paths. SIAM Journal on Computing, 28(2), 652-673.

Epstein, J. M., Pankajakshan, R., & Hammond, R. A. (2011). Combining computational fluid dynamics and agent-based modeling: A new approach to evacuation planning. PloS one, 6(5), e20139.

Fang, J., El-Tawil, S., & Aguirre, B. (2016b). Leader–follower model for agent-based simulation of social collective behavior during egress. Safety Science, 83, 40-47.

Fang, J., El-Tawil, S., & Aguirre, B. (2016a). New agent-based egress model allowing for social relationships. Journal of Computing in Civil Engineering, 30(4), 04015066.

Langford, W. P. (2010). A space-time flow optimization model for neighborhood evacuation. NAVAL POSTGRADUATE SCHOOL MONTEREY CA.