Computers, Environment and Urban Systems 80 (2020) 101452

Contents lists available at ScienceDirect

Computers, Environment and Urban Systems

journal homepage: www.elsevier.com/locate/ceus

ViCTS: A novel network partition algorithm for scalable agent-based modeling of mass evacuation

Dandong Yin[a,b], Shaowen Wang[a,b,*], Yanfeng Ouyang[c]

[a] Department of Geography and Geographic Information Science, University of Illinois at Urbana-Champaign, Urbana, IL, USA
[b] CyberGIS Center for Advanced Digital and Spatial Studies, University of Illinois at Urbana-Champaign, Urbana, IL, USA
[c] Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, USA

ARTICLE INFO

Keywords:
CyberGIS
Large-scale evacuation modeling
Network partition
Agent-based modeling

ABSTRACT

Emergency evacuation is a critical response to deadly disasters such as hurricanes, floods, and earthquakes, etc. However, mass emergency evacuation itself is a complex process that sometimes could lead to chaotic situations and unintended consequences. In many emergency scenarios, mass evacuation is necessary to cope with severe public threats within tight spatiotemporal ranges. To better understand complex phenomena like mass evacuation, and study possible consequences, agent-based models (ABMs) have been widely developed in previous work. Existing models simulate individual behaviors, posing computational challenges when applied to large geographic areas and sophisticated behaviors. A key strategy for resolving such computational challenges is to partition transportation networks into smaller regions and resolve corresponding computational costs by taking advantage of advanced cyberinfrastructure and cyberGIS. In this study, a novel network partition algorithm is developed to improve the scalability of agent-based modeling of mass evacuation based on a cutting-edge cyberGIS-enabled computational framework that exploits the spatial movement patterns of emergency evacuation. Specifically, the algorithm is termed as Voronoi Clustering based on Target-Shift, or ViCTS. It is enlightened by network Voronoi diagrams and designed to resolve computational scalability challenges caused by the unique characteristics of evacuation traffic. We conducted a set of computational experiments with real street network data in various evacuation scenarios to test the effectiveness and efficiency of the algorithm. Computational experiments show that ViCTS outperforms a widely used network partition algorithm for microscopic traffic simulation in terms of achieving optimal computational performance by balancing computational loads and reducing communications across high-performance parallel computing resources.

1. Introduction

Emergency evacuation is critical to saving lives and properties in the context of responding to deadly natural disasters such as hurricanes, floods, and earthquakes, etc. However, mass evacuation is a complex process sometimes leading to undesirable and chaotic outcome. Therefore, it is important to gain rigorous understanding of this process for designing effective evacuation strategies and assessing evacuation consequences. For this purpose, extensive studies have been conducted to understand the dynamics and consequences of emergency evacuation (Lu, George, & Shekhar, 2005; Murray-Tuite & Wolshon, 2013; Shekhar et al., 2012; Wolshon & McArdle, 2009).

One of the most significant challenges in understanding the emergency evacuation process is its complex nature. For example, a mass evacuation could involve a huge population; and each individual has a unique situation with different incentives and motives; meanwhile the behaviors and decisions of individuals tend to influence those of each other, leading to collective uncertain evacuation dynamics, which is hard to predict through analytical or empirical approaches (Fieguth, 2016; Fuchs, 2012; Lawler, Thye, & Yoon, 2014). Literature suggests computational modeling, in particular agent-based models (ABMs), as an effective way to represent such uncertain dynamics. Specifically, these modeling approaches represent collective interactions, synthesize heterogeneous data sources, and reveal implicit correlations (Chen, 2008; Farahmand, 1997; Han & Yuan, 2005; Suzumura, Houngkaew, & Kanezashi, 2014; Yuan et al., 2017). However, the computational intensity (Wang, 2008) of ABMs tends to increase dramatically as population size and related spatial scope expand. This computational intensity challenge often limits agent-based evacuation modeling to relatively small areas (Yan, 2014), which jeopardizes the purpose of

* Corresponding author at: Department of Geography and Geographic Information Science, University of Illinois at Urbana-Champaign, Room 2046, Natural History Building, 1301 W. Green St., Urbana, IL 61801, USA.
E-mail addresses: dyin4@illinois.edu (D. Yin), shaowen@illinois.edu (S. Wang), yfouyang@illinois.edu (Y. Ouyang).

https://doi.org/10.1016/j.compenvurbsys.2019.101452
Received 14 March 2019; Received in revised form 17 November 2019; Accepted 28 November 2019
Available online 28 December 2019
0198-9715/

ABMs for gaining holistic insights.

Meanwhile, with recent advances in cyberinfrastructure (Hey & Trefethen, 2005), enabling large-scale microscopic traffic simulation based on high-performance computing has been pursued extensively (Rickert & Nagel, 2001; Smith, Beckman, Anson, Nagel, & Williams, 1995; Suzumura et al., 2014; Yan, 2014; Wen, 2008). Such research efforts shed light on how to improve the computational scalability of agent-based traffic simulation. Nonetheless, gaps still exist for applying scalable traffic modeling and simulation to emergency evacuation. A major gap is that the unique characteristics of evacuation traffic (Dixit & Wolshon, 2014; Wolshon & McArdle, 2009) are not incorporated into the design of generic traffic simulation, causing low computational performance.

In this research, we analyze and demonstrate the inefficiency of directly applying the scalability strategy of generic traffic modeling to evacuation scenarios. To fill this gap, we propose a novel network partition algorithm for scalable agent-based evacuation modeling. The algorithm is termed as Voronoi Clustering based on Target-Shift (ViCTS). It is enlightened by network Voronoi diagrams (Okabe, Satoh, Furuta, Suzuki, & Okano, 2008) and specifically designed to resolve computational scalability challenges caused by the unique characteristics of evacuation traffic. We have conducted a set of computational experiments with real street network data in a specific evacuation model scenario to test the effectiveness and efficiency of the algorithm, as compared to the most popular counterpart of generic traffic modeling: METIS (Karypis & Kumar, 1995).

In the following sections, we first provide an overview of related work in scalable traffic modeling and identify the unique challenges of modeling evacuation traffic in a scalable fashion (Section 2). Subsequently, we describe our work that exploits the unique spatial patterns of evacuation traffic into consideration to resolve the corresponding computational challenges (Section 3). Finally, a series of computational experiments using our network partition algorithm against a mainstream method: METIS (Karypis, Aggarwal, Kumar, & Shekhar, 1999) is conducted on Miami's street network for a simulated traffic evacuation (Section 4). To our best knowledge, our work represents the first network partition solution for scalable evacuation emergency simulation.

2. Background

2.1. Network partitioning methods for scalable agent-based traffic modeling

ABMs have been demonstrated to be effective for simulating dynamic conditions and varying evacuee responses typically found in a mass evacuation (Chen, 2008; Chen, Meaker, & Zhan, 2006; Cova & Johnson, 2002; Farahmand, 1997; Han & Yuan, 2005; Richard & Church, 2002) However, the computational intensity of such models can be prohibitively high especially when the number of agents and corresponding spatial domain are large. For the agents, aside from computing resources needed to handle each one of them, there are also interactions among these agents and between the agents and their environments that increase dramatically as the number of agents grows. In a straightforward case of traffic modeling, a vehicle needs to know at least the positions of preceding vehicles all the time in order to avoid collision. For a spatial domain, when modeled in a sizable spatial network, even the best shortest-path algorithms pose serious computational challenges. In an evacuation scenario where road conditions vary dynamically, all vehicles modeled as agents need to update their shortest paths frequently in order to get their optimal paths. All of these processes need to be computed for each time step until the entire evacuation ends. Given the above challenges, it is practically infeasible to apply agent-based evacuation models to a large spatial domain such as at the scale of populous cities without computationally scalable models (Yan, 2014). Therefore, it is necessary to develop computational scalability strategies to exploit advanced cyberinfrastructure and high-performance computing for agent-based evacuation models.

To fully leverage cutting-edge cyberinfrastructure, a key strategy is to design an agent-based model in a scalable fashion. For example, TRANSIMS (Rickert & Nagel, 2001; Smith et al., 1995) uses a cellular automata model that divides roads into car-length cells, each of which can be occupied at most by one vehicle at the same time. The movement of vehicles is modeled as the transition of vehicles from one road cell to another. The distance of each transition represents the speed of the vehicle at that moment. In this design, road cells could be assigned to different logical processes (LPs) in parallel, with a message passing interface to communicate vehicle transitions between different LPs.

The fundamental idea of scaling up agent-based traffic simulation with network partitioning is illustrated in Fig. 1. Network partitioning decomposes the spatial domain into a set of sub-networks, which provides a basis to decompose the underlying computation into a group of processors. Each processor computes the movements of agents in its area of focus in parallel, while constantly communicating with neighboring processors when some agents go across their partition boundaries. The efficiency of parallelization is measured based on how balanced the computational work is split among processors (load-balance), as well as the frequency and volume of communication between processors (communication cost). (Rickert & Nagel, 2001) proposed a conceptual framework to evaluate the overall computational efficiency of scalable traffic modeling:

$$T_{parallel} = T_{cmp} + T_{cmm} \tag{1}$$

$$T_{cmp} = \frac{T_{seq}}{n}(1 + f_{overhead} + f_{load}) \tag{2}$$

$$T_{cmm} = n_{neighbor} \cdot T_{lat} + n_{message} \cdot T_{message} + \frac{C}{C_{sat}} \tag{3}$$

where $T_{cmp}$is the total time for computation; $T_{cmm}$is the total time of communication; T_{seq} is the sequential computation time; $f_{overhead}$is the overhead effects (e.g. maintaining copies of the common roads and vehicles on both LPs); $f_{load}$is the effect of load balance; $n_{neighbor}$is the number of neighboring LPs for one LP, $T_{lat}$is the latency (start-up time) of each message;$n_{message}$is the number of messages to be sent (size of communication); $T_{message}$is the unit time cost for sending one message; and $\frac{C}{C_{sat}}$accounts for the network saturation effects, which is dependent on the network topology.

Generally, as $f_{overhead}$ is related to the number of edge splits, which also contributes to part of the communication cost, network topology is

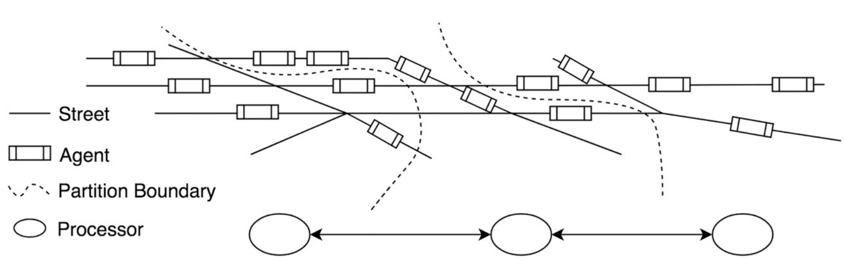

Fig. 1. Network partitioning for scalable agent-based traffic simulation.

costly to optimize. The major goals of optimizing total computational efficiency are to control 1) load balance and 2) communication cost, both of which are critically affected by the mapping from traffic network to LPs through spatial domain decomposition and network partitioning.

Based on the goal of computational performance optimization, the following types of network partitioning methods have been studied (Xu, Cai, Aydt, & Lees, 2014):

1) **Geographic partitioning** (Klefstad, Zhang, Lai, Jayakrishnan, & Lavanya, 2005; Sun, Chen, Li, & Wang, 2012; Wei, Chen, & Sun, 2010) decomposes a street network directly using geographic locations of roads and intersections. Apparently, this kind of partitioning methods achieve optimal performance when the computational load and communication distribution have explicit and predictable geographical patterns. For example, police vehicles patrol in their own jurisdiction. However, this assumption does not hold for most complicated traffic scenarios, such as in emergency evacuation. Due to its marginal computational cost, it is usually adopted for dynamic repartitioning where the computational speed of partitioning is preferred over optimality.
2) **Scattered partitioning** (Barceló et al., 1998; Thulasidasan, Kasiviswanathan, Eidenbenz, & Romero, 2010) aims for load balance by dividing road networks into small and scattered parts, regardless of their spatial proximity and connectivity. Consequently, this type of partitioning usually induces significant communication cost. Therefore, it is mainly used when the cost of communication is marginal, which can be achieved for example through shared-memory architecture.
3) **Graph partitioning** (Fiduccia & Mattheyses, 1988; Hendrickson & Kolda, 2000; Kernighan & Lin, 1970; Xu, Cai, Eckhoff, Nair, & Knoll, 2017) formulates the network partitioning problem as a graph partitioning problem by converting a road network into a graph and assigning workload indices as weights to corresponding vertices and edges in the graph. Graph partitioning is generally flexible and comprehensive to represent different trade-offs between load-balance and communication. The major disadvantage is that many graph partitioning methods are computationally intensive. Therefore, hypergraph partitioning gains popularity among graph partitioning methods due to its relatively low computational cost and reasonable optimality. In hypergraph partitioning, the target graph is generalized to a hypergraph by grouping parts of the graph (subgraphs) as hyper vertices connected by hyper edges, which reduces the partition problem to a smaller scale and reduces computational cost of partitioning. In practice, the *k*-way multi-level Kernighan-Lin hypergraph partitioning method of METIS and its variants (Karypis et al., 1999; Karypis & Kumar, 1995; Karypis, Schloegel, & Kumar, 1997; Xu & Tan, 2012) are widely adopted for scalable agent-based traffic modeling.

However, most existing network partitioning methods, even those widely used for scalable traffic modeling, do not adequately take the spatial patterns of traffic flows into consideration. Many graph partition methods including METIS, mainly optimize the edge cuts, load-balance, or a combination of both in their partitioning algorithms. While such optimization is reasonable for most networks, and practically sufficient for generic traffic modeling, it is not optimal when the traffic flows have distinct spatial patterns such as those observed during an emergency evacuation. In the next section, we first summarize the unique patterns of evacuation traffic, and then analyze why such patterns can cause suboptimal performance of existing network partition methods.

2.2. Evacuation traffic modeling

As summarized by (Gan, Richter, Shi, & Winter, 2016), evacuation is "a time critical process in which the highest priority is to get those people who may be affected by a disaster out of the danger zone as fast as possible." Undoubtedly, this kind of transportation demand (moving out of a region under time pressure) is the most distinguishable characteristic that is rarely observed or modeled for generic traffic. Besides the general demand of moving outwards, there are also specific demands of urgency and route planning affected by various factors, including demographic, socio-economic, cultural and personal differences of each household and individual, as well as local conditions and the development of disaster threats (Murray-Tuite & Wolshon, 2013; Yin, Murray-Tuite, Ukkusuri, & Gladwin, 2014; Yuan et al., 2017).

Driven by this demand, unique patterns of evacuation traffic have been investigated by a series of empirical studies (Dixit & Wolshon, 2014; Wolshon & McArdle, 2009). For example, it is widely observed that regardless of any physical locations or events, the capacity of traffic flows in evacuation inevitably drops by 10–20% after reaching a peak during early evacuation phases, and then the dropped capacity is sustained for periods of six to twelve hours (Banks, 1990; Dixit & Wolshon, 2014; Wolshon, 2008). This phenomenon, referred as capacity drop, can be caused by congestions in downstream queues (Brilon, Geistefeldt, & Regler, 2005; Tu, van Rij, Henkens, & Heikoop, 2010). The capacity drop phenomenon is an example of how the supply side of traffic modeling can be volatile in an evacuation scenario. Events like traffic congestion, traffic accidents, or road damage caused by disasters can all result in dramatic drop of overall network capacity, potentially leading to massive re-navigation and complex traffic dynamics.

In general, we summarize the unique characteristics of evacuation traffic as compared to normal traffic as follows:

- **Spatially heterogeneous demand:** Evacuation traffic has clearly heterogeneous demands (moving outwards from danger zones) and traffic flows are non-uniform (diffusing towards the nearest exits).
- **Dynamically uncertain supply:** Road conditions are subject to irregular events that reduce the capacity (supply) of the whole network in the middle of evacuation, and force vehicles to re-navigate their routes dynamically.

The spatial heterogeneity and dynamic uncertainty of evacuation traffic lead to significant computational challenges that traditional network partitioning methods are ill-suited to address. First of all, spatially heterogeneous demands could lead to severe load imbalance for traditional network partitioning based on homogeneous assumptions. Regardless of the initial vehicle distribution and evacuation efficiency, as long as traffic flows are moving out of danger zones in general, vehicles will naturally concentrate on the periphery or outside of danger zones and dilute in centers as the evacuation process unfolds. If the network partitioning does not take this into consideration and assumes homogeneous vehicle distributions over the entire evacuation duration, it is likely that the center partition will be idle while the peripheral partitions are severely swamped, which could cause significant load imbalance and computation inefficiency.

However, the challenge posed by spatial heterogeneity alone is not intractable. As long as the demand pattern is stable, corresponding evacuation routes can be pre-computed to reveal the traffic load distribution across the entire street network. Then it becomes a typical network partitioning problem with certain weight distribution over network nodes and edges, which is well addressed by traditional partitioning algorithms (Karypis & Kumar, 1995). What makes the problem more difficult is the combination of spatially heterogeneous demand and dynamic uncertain supply. While congestion-induced supply reduction is still possibly manageable and predictable, supply reduction caused by accidents or disaster damage is extremely complicated to model and has to be represented by randomness. This means that there will be many vehicles dynamically updating their evacuation routes and possibly changing their target exits ("horizontal" movements) in an unpredictable fashion. Therefore, even if traditional network partitioning is designed to pay extra attention to peripheral exits/shelters, without properly handling the dynamic supply drop and the corresponding traffic rerouting, the overall computational performance could still suffer from large communication volume from vehicle transitions across partition boundaries.

3. Method

Previous analysis has shown that the spatial heterogeneity and uncertain supply of evacuation traffic have to be taken into consideration for optimal computational performance, and that it is difficult to incorporate these characteristics into traditional methods. In this section, we propose a new network partition algorithm using Voronoi-Clustering based on Target-Shift (ViCTS) to solve the aforementioned problems.

3.1. Network Voronoi diagrams for evacuation

Instead of partitioning the network into arbitrary granularity, we build final partitions from atomic parts, specifically network Voronoi diagrams (Okabe et al., 2008), to address the spatial heterogeneity problem. The detailed mechanism is explained in the following proof and discussion.

Network Voronoi diagrams extend the traditional planar Voronoi diagrams (Voronoi, 1908) to network space, by

- Restricting the spatial domain of interest from the entire geometric plane to the edges and vertices on a specific network; and
- Replacing the distance definition from planar distances (Euclidean distance, Manhattan distance, etc.) to the network distance (length of the shortest path between two points on the network).

The formal definition of network Voronoi diagrams is as follows: given a network graph $G = (V,E)$ and a set of seed vertices $S = \{s_1 \ldots s_n\}$ on G, the network Voronoi diagram $\{VI_1 \ldots VI_n\}$is defined by:

$$VI_i = \{v \mid d_G(v,s_i) \leq d_G(v,s_j), j \neq i, j = 1..n, v \in V\} \tag{4}$$

where $d_G(v_1,v_2)$ is the network distance from one vertex to another. It should be noted that the distance could be directional depending on the directionality of G. In the case that the distance is directional, what is defined by Eq. (4) is actually an *inward* Voronoi diagram (Okabe et al., 2008), which is used in this study for solving the network partitioning problem.

In an evacuation scenario, we define the Voronoi diagram as specified in (4), using the street network as G; the set of safe exits (i.e. road intersections outside the periphery of the danger zone) as the seed vertices; and the travel distance as network distance (see Fig. 2 for an illustrative example). This way we can observe the following evacuation pattern for all vehicles initially located in a Voronoi subnet VI_i:

- All the vehicles in this subnet have the same nearest exit s_i, hence they are likely to choose that s_i as the same destination for evacuation;
- The shortest path from their initial locations to$s_i$completely falls in $\overline{VI_i} = VI_i + \partial VI_i$, i.e.

$$\forall\ v \in \overline{VI_i}\ and\ p(v,s_i) = \{v -> v_1, v_1 -> v_2, \ldots, v_m -> s_i\}\ with\ length(p(v,s_i)) = d_G(v,s_i),\ \forall\ j \in \{1..m\},\ v_j \in \overline{VI_i} \tag{5}$$

Here ∂VI_i stands for those vertices that have equal distance to s_i and another s_j, $\partial VI_i = \{v \mid \exists j, d_G(v,s_i) = d_G(v,s_j)\}$, which technically belongs to both Voronoi subnets. But in reality, with road segment lengths measured as floating-point numbers, the chance for one origin to have exactly the same shortest distance to multiple destinations is extremely small, hence most of the time $\partial VI_i = \phi$ and $\overline{VI_i} = VI_i$.

(5) can be proved by contradiction: if a shortest path $p(v,s_i)$ lands on a vertex $v_k \notin \overline{VI_i}$which belongs to another Voronoi subnetwork VI_j, then we have

$$d_G(v,s_i) = d_G(v,v_k) + d_G(v_k,s_i) \tag{6}$$

Since $v_k \notin VI_j$ and $v_k \in VI_j$, according to the definitions of VI_j and$\overline{VI_j}$,

$$d_G(v_k,s_i) > d_G(v_k,s_j) \tag{7}$$

Combining (6) and (7),

$$d_G(v,s_i) = d_G(v,v_k) + d_G(v_k,s_i) > d_G(v,v_k) + d_G(v_k,s_j) = d_G(v,s_j) \tag{8}$$

meaning $v$is strictly closer to $s_j$thans_i, which contradicts the definition of VI_iin (4) that requires $d_G(v,s_i) \leq d_G(v,s_j)$.

In summary, if all vehicles in the same Voronoi subnet choose to evacuate to the nearest exit, not only will they move towards the same destination, but also all of their routes will completely fall in the range of that Voronoi subnet. This is a strong indication for network partitioning: if a partition consists of a set of complete Voronoi subnets, then all vehicles in this partition will stay within the partition during the entire evacuation, and hence vehicle transitions across partitions will be zero. However, this is only the case when every vehicle holds to their shortest path, and road conditions remain constant.

3.2. Road closure and target shift

While network Voronoi diagrams are effective for solving the partitioning problem in static cases, it remains difficult to deal with dynamic, random road closures during the evacuation. To assess this difficulty, the consequences of road closures for evacuation traffic need to be analyzed. However, in a real -world mass evacuation, these consequences could be complex, depending on how the road closure information is disseminated and how each individual reacts to the situation. For the sake of simplicity and the computational focus of this research, we assume that information dissemination is complete and instant, and all evacuation vehicles make decisions in the following pattern:

- If the closed road is not on the shortest route to a vehicle's current destination, then this vehicle is not affected and will resume its travel as normal.
- Otherwise, if the shortest route is now partially closed, then:
  - If a short detour exists (the second-shortest path towards the same destination), the vehicle should take that detour and stick to the previous destination
  - If the road closure is critical (the path towards the current destination increases tremendously), then the vehicle should re-navigate to the current nearest exit, given the new road conditions.

As long as the vehicle sticks to its previous destination, it is not likely to detour very far and hence would possibly remain geographically close to its Voronoi subnet. The real obstacle here is the kind of road closure that leads vehicles to switch to other exits, which could result in dramatic changes in the vehicles' routes. To tackle this problem, we need an effective indicator to predict the range of alternative destinations a vehicle would choose in case of road closures (target shift).

Intuitively, the more numbers of roads between a vehicle and an exit, the less likely the vehicle is going to shift its target. Specifically, assuming the closure of roads is subject to an identical independent distribution (iid), then the “stickiness” between a vehicle's location and an exit could be measured by the minimum number of road segments that have to be closed in order to increase the shortest distance to be larger than the distance between the alternative exit (second-nearest) and the vehicle and its current location. Formally, if we define this number as M, then:

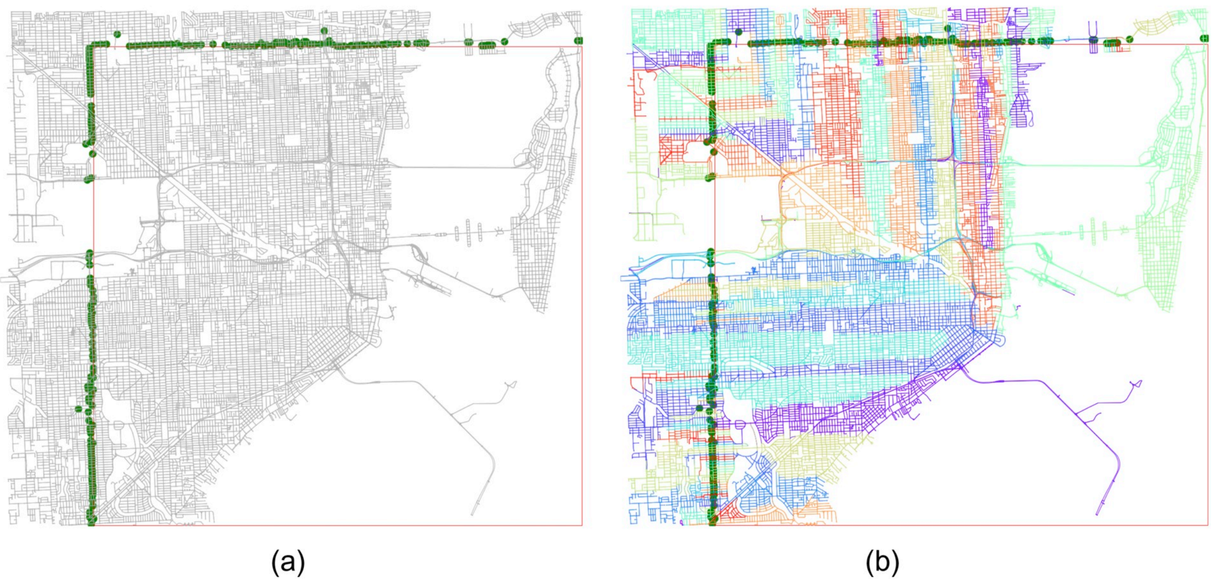

Fig. 2. (a) Miami road network with hypothetical evacuation sections area (red rectangle) and exits (green dots) (b) Miami road network Voronoi indicated by color. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$M(L, v, s_i) = min|\{c \mid d_{G-c}(v, s_i) > L\}| \quad (9)$$

where L is the length limit, which could be the value of $d_G(v, s_j)$, wheres_j stands for the second nearest exit of v, and c is a certain set of edge cuts or road closures. In general, to define the minimum number of edges to be cut for a vehicle to choose the n^{th}closest exit $s_v{}^n$as its destination, we have

$$C(v, n) = \sum_{i=1}^{n-1} M(d_G(v, s_v^n), v, s_i) \quad (10)$$

Theoretically, a minimum number could correspond to multiple road closure situations. However, in a realistic network, especially when road lengths are measured as float numbers, the minimum solution is practically unique. Therefore, given the identical independent distribution (iid) of each road closure, the probability for a vehicle to choose its n^{th} closest exit as an evacuation destination, with the least number of roads being closed equals to the multiplication of each individual road closure:

$$P(v, n) = \left(\frac{r}{|E|}\right)^{C(v,n)} = r^{C(v,n)} \cdot |E|^{-C(v,n)} \quad (11)$$

where r is the probability for any road to be closed, and $|E|$ is the number of roads in the entire region. Now we need to find out how $P(v, n)$ unfolds for different v and n.

Unfortunately, there is a theoretical roadblock in the above deduction: the computation of $M(L, v, s_i)$ defined in (9) is known as the *minimum length-bounded edge cut* problem, whose general solution has been proven *NP*-hard to approximate even in unit-edge-length graphs (Baier et al., 2006). Therefore, the exact value of $C(v, n)$and $P(v, n)$ will be impractical to compute for any sizable road network, and thus we seek an effective approximation instead.

What (Baier et al., 2006) pointed out is that in general, for any theoretically possible origin-destination pair (v, s_i) on any graph G, $M(L, v, s_i)$ is expected to grow approximately exponentially with respect to L, i.e.

$$M(L, v, s_i) \sim k^L \quad (12)$$

where $k > 1$ is a constant. Therefore,

$$C(v, n) = \sum_{i=1}^{n-1} M(d_G(v, s_v^n), v, s_i) \sim \sum_{i=1}^{n-1} k^{d_G(v, s_v^n)} = (n-1)k^{d_G(v, s_v^n)} \quad (13)$$

and

$$P(v, n) = \left(\frac{r}{|E|}\right)^{C(v,n)} \sim \left(\frac{r}{|E|}\right)^{(n-1)k^{d_G(v, s_v^n)}} \quad (14)$$

Now the key component is $d_G(v, s_v{}^n)$, whose exact value is highly dependent on the actual location of the vertex v, the graph G, and the set of exits $\{s_i\}$, but nonetheless is computable in polynomial time using any shortest path algorithms. In general, as $d_G(v, s_v{}^n)$ is the exponential part of an exponential; its increase will be dramatically amplified towards $P(v, n)$, meaning that the probability for a vehicle to take faraway exits decreases extremely quickly. Even if in edge cases where v is in the center of all exits $S = \{s_i\}$ and $d_G(v, s_v{}^n)$ remains constant with respect to n, then $P(v, n)$ still decreases exponentially with respect to n.

$$P(v, n) \sim \left(\frac{r}{|E|}\right)^{(n-1)k^{d_G(v, s_v^n)}} > \left(\frac{r}{|E|}\right)^{k(n-1)} \quad (15)$$

In conclusion, the above formulation captures the target-shifting behavior caused by random road closures, which can be leveraged to achieve optimal network partitioning. Generally, it is appropriate to assume that a vehicle is most likely to switch among a few of its nearest exits and disregard distant ones. Depending on the specific needs and available information, either (14) or (15) could be used as an estimation of the target-shifting probability.

We conducted a set of computational experiments on real street network data (see Section 5) to further validate the above conclusion. Fig. 3 illustrates the percentage of vehicles that finally evacuated at their n^{th} nearest exit based on a set of simulations with different rates of random road closures. For example, 0.005/30 means that every road has a 0.5% chance to be closed in every 30 s (which means on average about 55% of all roads in a city will be closed in an hour). As indicated by the formulation, most vehicles will end up at a small number of closest exits, which provides quantitative support for defining the Target-Shift proximity.

The Target-Shift proximity between a pair of sub-graphs $G_1$and G_2 of G is defined as follows:

$$TS(G_1, G_2) = \sum_{v \in V_1} \sum_{s \in S_2} P(v, n(v, s)) + \sum_{v \in V_2} \sum_{s \in S_1} P(v, n(v, s)) \quad (16)$$

where $V_1$and V_2 are the set of vehicles located in G_1 and G_2 respectively; $S_1 = S \cap G_1$ and $S_2 = S \cap G_2$ are the set of exits in the corresponding subgraphs; $n(v, s)$ is the distance rank of an exit to an vehicle, i.e. s is the

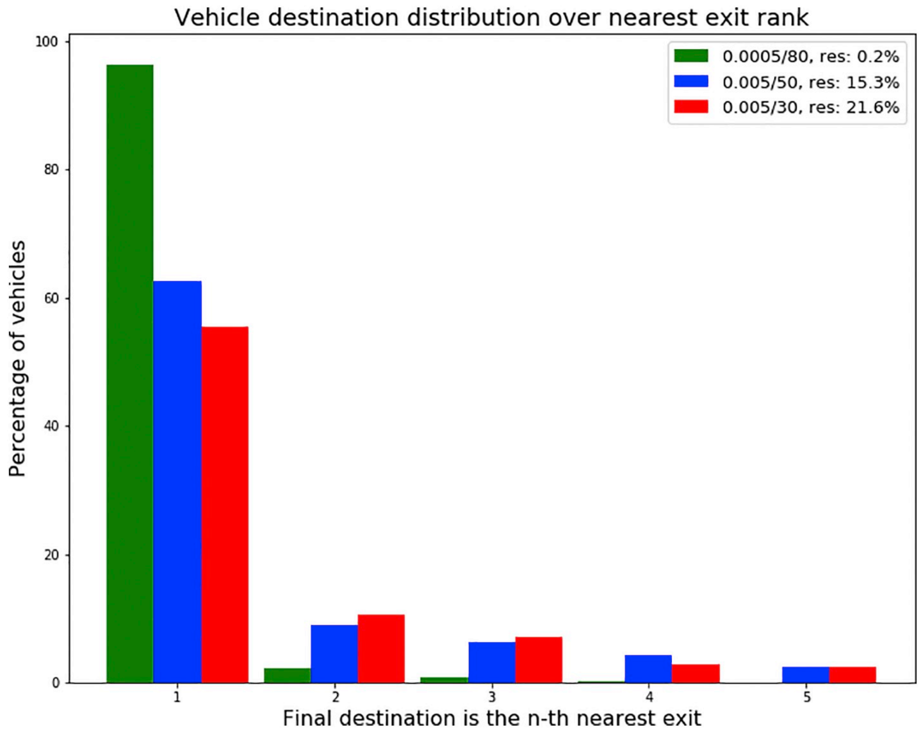

Fig. 3. Distribution of vehicles' final destinations in the case of random road closure.

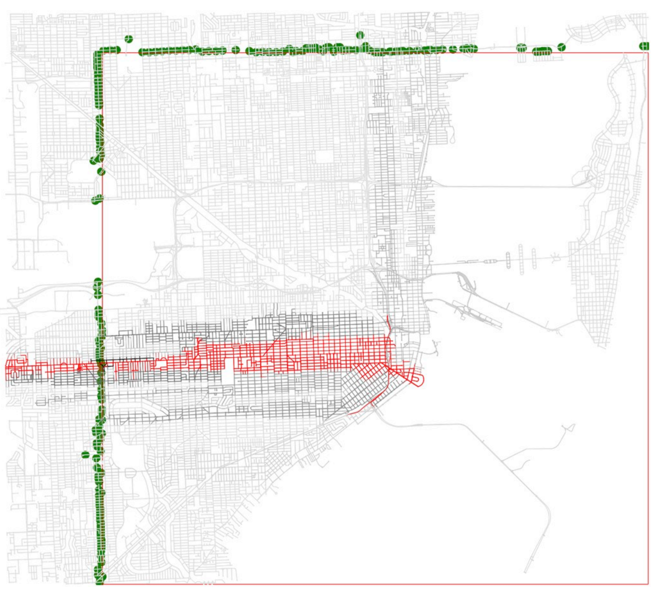

Fig. 4. The Target-Shift proximity for an example Voronoi area (red) to other Voronoi areas (darker color indicates closer distance in terms of Target-Shift (TS) proximity). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

n-th nearest exit for v. The general idea of (16) is to aggregate the probability of all possible vehicle transitions between the two subgraphs. In an actual simulation, $P(v,n)$ could be estimated using either (14) or (15). A visual illustration of target-shift distance is shown in Fig. 4.

With the proof for the Voronoi subnets as a valid basis for partitioning, and the definition of TS-proximity between Voronoi subnets to represent the uncertainty of road closures, our network partitioning method for evacuation traffic modeling can be formulated by clustering Voronoi subnets using TS-proximity. Such clustering could vary, as it has to be traded off with load-balance. Since we have proven that partitioning based on Voronoi groups with short TS-proximity can reduce vehicle transitions across partitions, then the runtime load balance is likely to stay close to its initial value. Therefore, an effective clustering strategy should produce a load balanced result on top of the Voronoi subnets and TS-proximity.

In fact, if we consider the Voronoi subnets as abstract vertices, and the TS-proximity as weights of edges between each pair of Voronoi subnets, then the problem is transformed to a classical graph partitioning problem on the complete Voronoi graph. The reason we refer to this problem as "clustering" is to differentiate it from the original network partitioning problem, and also to emphasize the idea of "grouping" Voronoi subnets into larger units. Since this Voronoi-based graph is significantly smaller in size compared to the original road network, its partitioning can be solved using many classic graph partitioning methods, such as Kernighan-Lin (Kernighan & Lin, 1970) or Fiduccia-Mattheyses (Fiduccia & Mattheyses, 1988). Our computational experiments use Kernighan-Lin algorithm for Voronoi subnets clustering in favor of its explicit control over the final number of partitions.

3.3. Assumptions and limitations

To summarize this section, we propose and prove that using Voronoi subnets as basic units to form partitions for evacuation traffic modeling is valid and potentially efficient, and that target-shift proximity clusters Voronoi subnets for partitioning, in terms of reducing possible vehicle transitions against dynamic and uncertain road closures. Nonetheless, the effectiveness of our method is built upon a set of assumptions. In this subsection, we will discuss these underlying assumptions, assess possible problems in case these assumptions do not hold, and provide corresponding remediation for such cases.

For example, since the Voronoi subnets are defined by the set of exit points, in order to provide an adequate range of choices to form partitions from Voronoi subnets, the number of exits has to be significantly larger than the required number of partitions. In other words, the number of partitions generated by the proposed method will be no larger than the number of exits of the evacuation. This may cause some problems in the following scenarios.

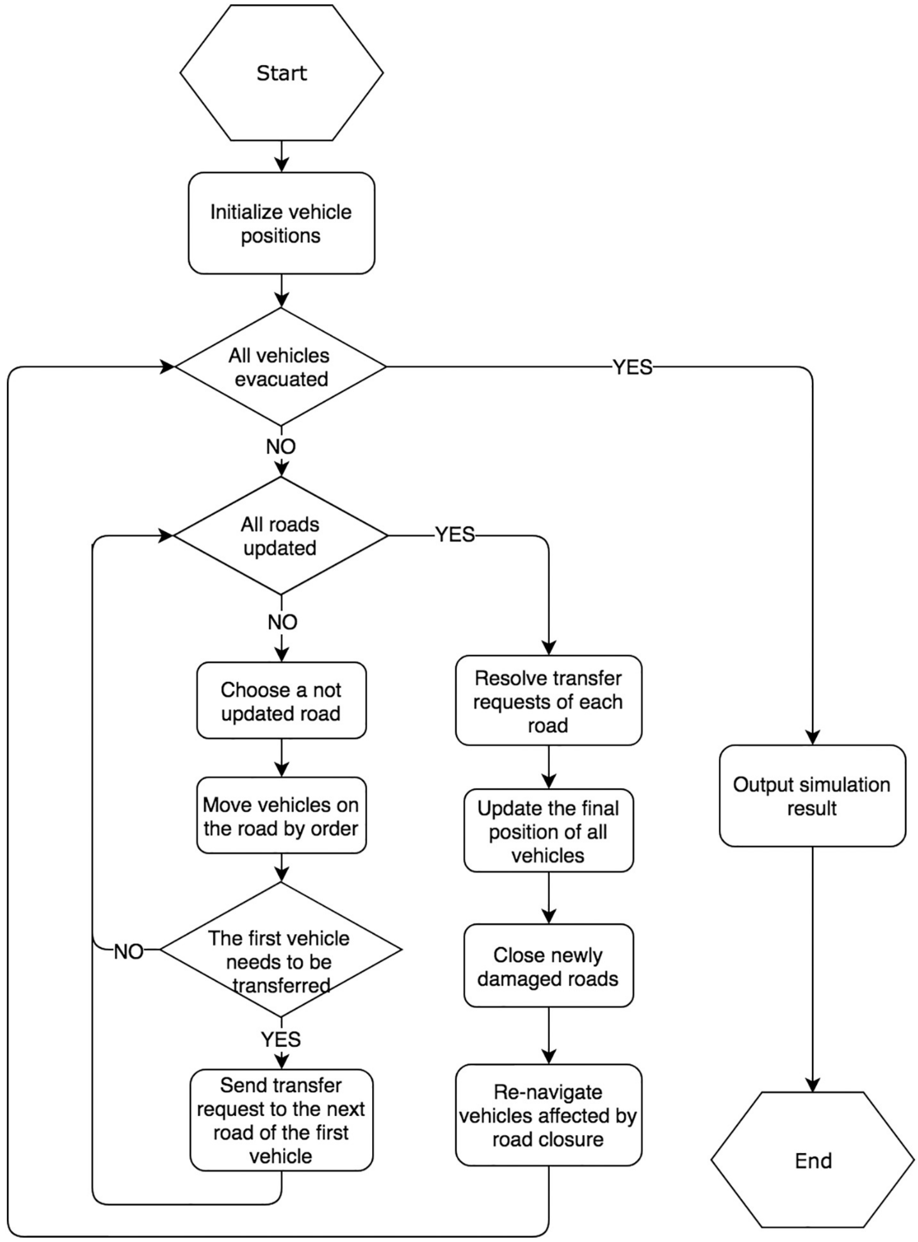

Fig. 5. Flowchart for the executing evacuation traffic model.

- When there are few exits available for a large population to evacuate through, e.g. when the danger zone is an island and the only exit is through a bridge. In this case, the proposed method could lead to an inevitably large load for each partition, and result in computational inefficiency. However, leaving aside tremendous social panic, in this scenario, the evacuation problem at the exits degenerates to simply traffic control at a few roads, which does not require a complicated model. Meanwhile, internal traffic, i.e. how to move vehicles efficiently to somewhere near these exits, is a more meaningful problem. Therefore, in this scenario, it is better to redefine (probably shrink) the evacuation zone to a more meaningful range before applying the proposed method.
- When there are huge amounts of computational resources to exploit, e.g. millions of CPUs on a supercomputer. As the number of partitions produced by the proposed method is limited by the number of exits, its scalability is bounded. However, we argue that the scalability of network-partition-based traffic simulation models are intrinsically bounded. Regardless of any particular partitioning methods, as the number of partitions increases, the average network size of each partition will inevitably be reduced. When the size of partitions becomes too small to host a meaningful number of vehicles for a substantial amount of time, vehicle transitions between partitions will be enormous, which will lead to overwhelming communication cost and inefficient overall computation. Therefore, if extreme scalability is desired, it is better to use other parallelization paradigms if possible.

Another underlying assumption is that the load for each Voronoi subnet is moderate. An exceptional case is where a single Voronoi subnet hosts a majority of the vehicles in the evacuation region. In this case, the proposed method might have difficulty to balance the roads among partitions, since it uses Voronoi subnets as basic units. However, still leaving aside the tremendous social consequences of such a scenario, the problem itself is not difficult to accommodate. In the presence of giant Voronoi subnets that cause severe load imbalance, it is fair to assume that not all vehicles in these Voronoi subnets plan to evacuate

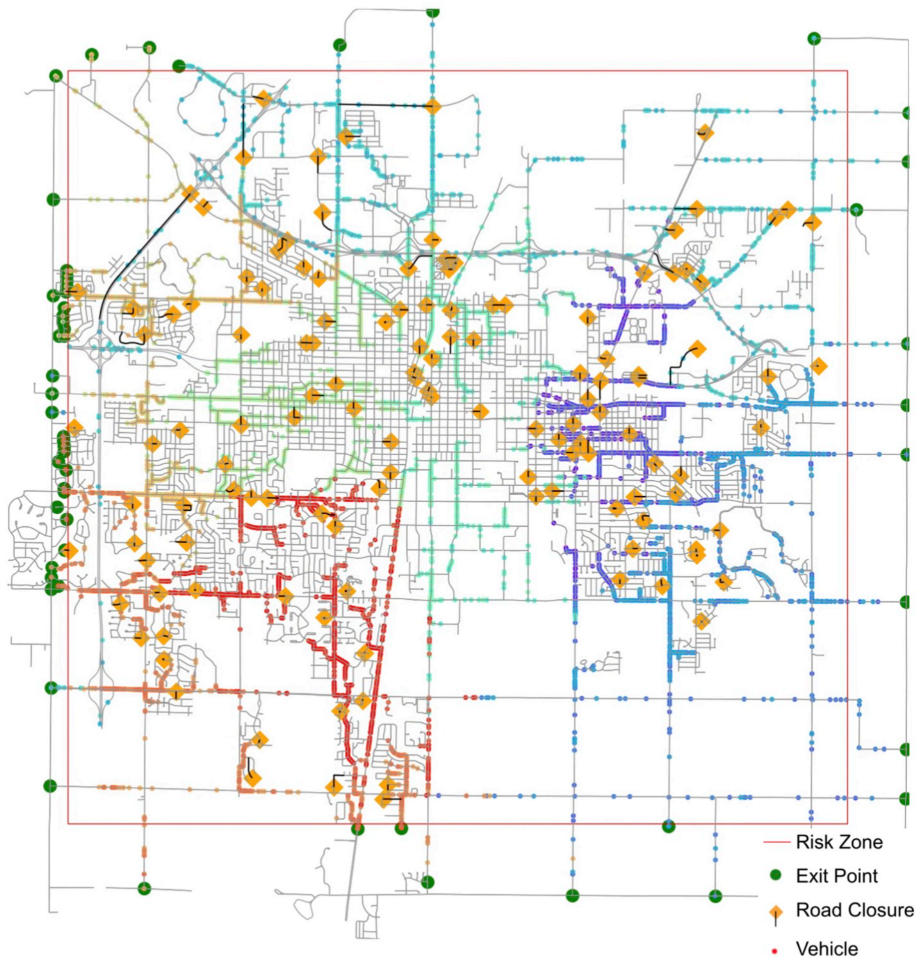

Fig. 6. A snapshot of a traffic evacuation simulation in the Champaign-Urbana area.

using their closest exits. Therefore, it is meaningful in reality and computation to pre-navigate some of the vehicles to nearby exits, which could be their 2nd, 3rd closest exits, etc. This way the giant Voronoi subnets are further broken down to smaller partitions based on the new destinations of the vehicles. Load balance could be achieved again using the split partitions. In case that re-navigation is forbidden and all vehicles must choose their closest exits to evacuate, the situation is then similar to the "few exits" case discussed previously, and the same logic applies.

4. Results

To evaluate ViCTS, we have conducted a series of computational experiments using real street network data and compare ViCTS against the widely adopted graph partitioning method for traffic simulation: METIS (Karypis et al., 1999). Results shown that ViCTS outperforms METIS for evacuation traffic modeling with superior computational performance.

4.1. Study area and data source

We chose the city of Miami as our study area for the following two reasons.

1. It is known for its high risk of flooding, which has led to several mass evacuations (e.g. the recent mass evacuation caused by Hurricane Irma in 2017).
2. It has a adequate diversity of terrain and transportation conditions including coasts, islands, bridges, as well as common metropolitan road networks with major highways.

We retrieved street network data for the city of Miami from OpenStreetMap (Haklay & Weber, 2008) using OSMnx (Boeing, 2017) with the bounding box: 80.316665° W, 25.703935° N, 80.119601° W, 25.858107° E (17 × 20 km). The retrieved street network contains 16,149 intersections and 46,707 road links, with an accumulated road length of 5638.83 km. The area is shown in Fig. 5.

The retrieved data is in a network structure with complete geometric attributes for each road lane. One-way roads are represented as directional edges in the network. However, due to the crowd-sourced nature of OpenStreetMap, the quality and coverage of the data is not uniform. For example, some traffic information, like speed-limits, is not available for all roads, and the traffic signal information is also incomplete. Therefore, we manually assign missing speed-limits values based on estimated values, and apply simple traffic control mechanisms, like stop-sign rules, to all intersections. However, it is also fair to assume weaker traffic regulations in an actual case of massive emergency evacuation. For the initial vehicle locations, we synthesize a location dataset of 500,000 vehicles. The number of vehicles is based on the census population estimation of Miami in 2017, which is about 463,000 (U.S. Census Bureau, n.d.). To increase the occurrence of vehicle transitions and target-shifts as a means to highlight scalability challenges, all vehicles are randomly distributed on every road.

4.2. Model specification and computing environment

(Gan et al., 2016) has formulated a conceptual evacuation model for optimizing staged evacuation with static routing. We use some of their configurations, but adapt for a microscopic, dynamic scenario with road closures and re-navigations specified as follows.

1. The disaster is still ongoing in an area of high risk during the evacuation; road infrastructure inside the region is subject to periodic failure (closure) with a certain probability;
2. All intersections outside the area of high risky region are considered an exit;
3. The exits are uncapacitated, i.e. evacuees reaching the exits are considered safe and thereafter removed from the model;
4. All evacuees start their evacuation processes at the same time;
5. Evacuees have instant global knowledge and always take the shortest path towards the nearest exit at any time (immediately respond to the road closures);
6. Evacuees drive either at the speed limit of the current road, or following the leading vehicle;
7. All intersections execute stop-sign rules (first come, first serve).

The workflow of the model is illustrated in Fig. 5.

The above model is implemented on the CyberGIS-Jupyter platform (Yin et al., n.d.) with the Python language. Fig. 6 shows a snapshot of an example evacuation of a test area in Champaign-Urbana (Illinois, USA). This test area is chosen as a small but complete road network that could clearly present the distribution of individual vehicles on roads with different topology and density. The red rectangle represents an area of high risk where roads could be closed (the orange diamonds) at any time. Large green circles indicate safe exits/shelters. Small dots on the roads are individual vehicles colored by the partition they belong to. There are 8000 vehicles in this simulation.

All computation work was conducted in a virtual machine environment with 10 Intel Xeon CPUs (2.40GHz), 50GB of RAM, Ubuntu 16.04, and Message Passing Interface (MPI) version 3.2.

4.3. Experiments and results

To test the efficiency of ViCTS in scalable computation, we have conducted two sets of computational experiments using the Miami data and compared the results against the most well-adopted graph partitioning method for traffic simulation: METIS (Karypis et al., 1999). The 2 sets of experiments differ in the vehicle routing/re-routing behavior. In the first set of experiments, vehicles always choose the path with the shortest distance to an exit as their evacuation route. In the second set of experiment, vehicles choose their paths with the shortest total estimated travel time. Simulation results show that ViCTS outperforms METIS for evacuation traffic modeling on load balance and communication volume in both sets of experiments.

We applied METIS and ViCTS to the study area and conducted evacuation simulation based on their partition results using a different number of CPUs. The performance of the simulation is evaluated based on the following two metrics:

1. Communication Cost: For each time step (one second), find the maximum number of vehicle transitions between all pairs of different CPUs, and compute the average transition volume during the entire evacuation process. Higher communication cost indicates higher networking latency leading to worse computational performance.
2. Load Balance: For each time step, find the partition with the least number of vehicles and the one with the most, calculate the ratio $\frac{min(t)}{max(t)}$ and compute the average ratio for all t. The overall load balance is a value in $[0, 1]$, where 0 means extreme imbalance and 1 means perfect balance.

The partition result is shown in Fig. 7. It is shown that METIS assigns the center of the evacuation zone to one partition (purple), while ViCTS divides that area into different partitions. While evacuating, vehicles in the central areas are likely to scatter along different directions, which could lead to significant cross-boundary movements in the METIS partition. Furthermore, METIS breaks possible evacuation paths into different partitions. For example, many paths from the center to the north or northwest fall into 3 partitions (purple, yellow, and red); most paths from the center to the west are cut into purple and cyan partitions. All of these lead to extra cross-boundary movements in evacuation, which increases the communication cost, and jeopardizes load balance.

Fig. 8 shows the actual performance metrics of ViCTS versus METIS in the shortest distance scenario. It is clear that ViCTS outperforms METIS in both metrics, which indicates better scalability.

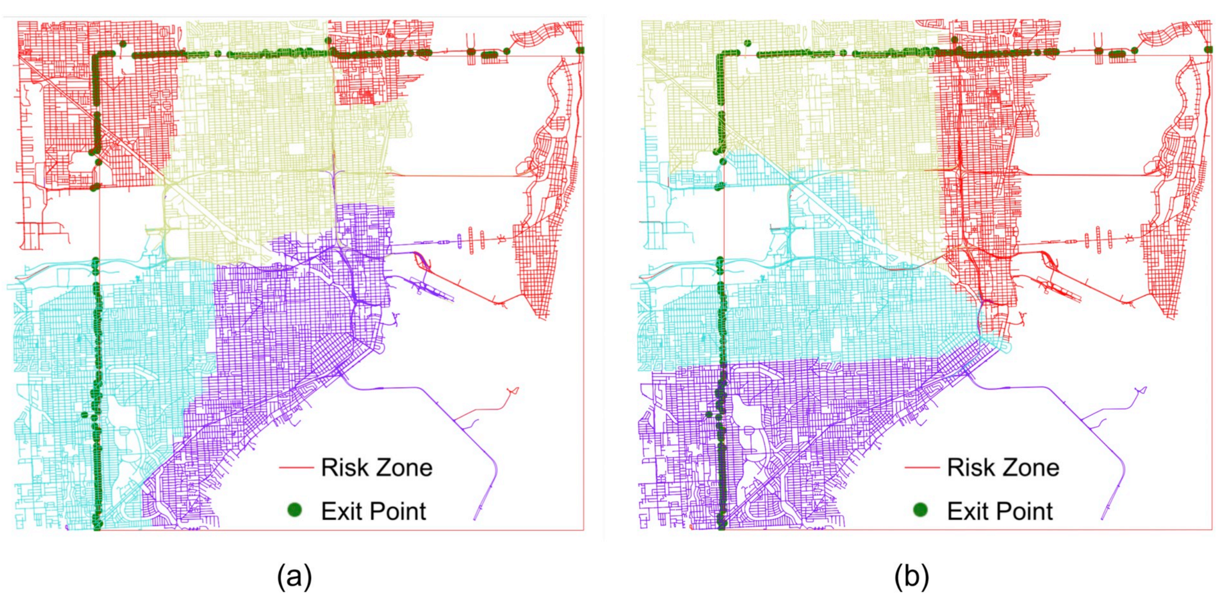

Fig. 7. (a) METIS's partition result on Miami road network (4 partitions, shown in different colors); (b) ViCTS' result on the same area (4 partitions, shown in different colors);

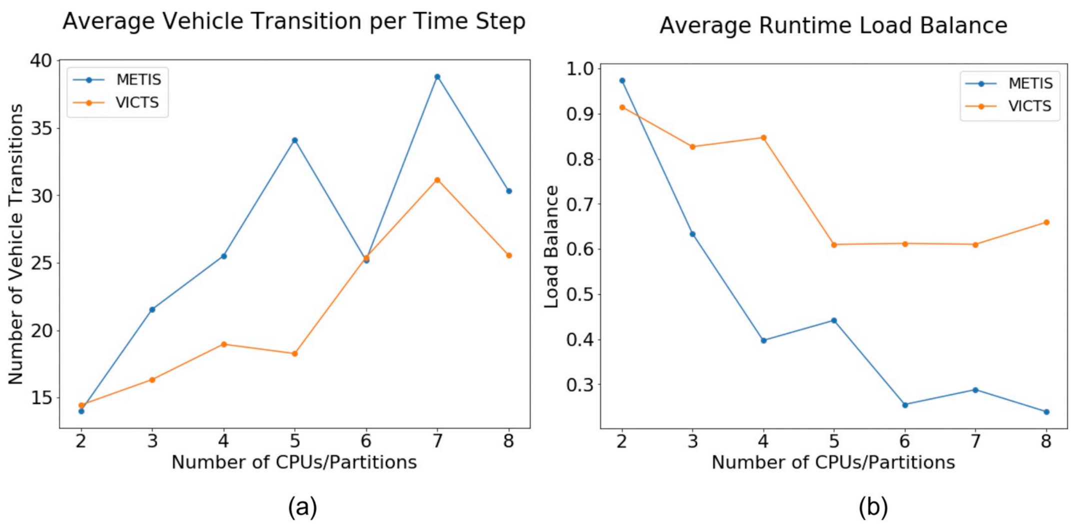

Fig. 8. (a) Communication cost of ViCTS and METIS for different partitions (lower is better); (b) Load balance of ViCTS and METIS for different partitions (higher is better). All experiments were conducted with a closure configuration of 0.005/30.

With the increasing availability of navigation services based on real-time traffic condition, it is reasonable to assume individuals plan their evacuation routes based on the real-time estimated travel time, instead of shortest distance (Miller-Hooks, 2001; Sheffi, Mahmassani, & Powell, 1982). To further test the efficiency of ViCTS, we conducted another set of experiments with the exactly same parameters except the individual routing behavior.

In this configuration, individuals are set to choose their evacuation route as the one with the shortest estimated travel time. The estimated travel time of each road is obtained as follows:

1) If there are no vehicle traveling on this road:

$$\textit{Estimated Travel Time} = \textit{Road Length}/\textit{Speed Limit} \tag{17}$$

The speed limit data are retrieved from OpenStreetMap.

2) Otherwise, assuming there are n vehicles on this road, each vehicle v_i has traveled a distance of d_i within the time of t_i since it enters the road.

$$\textit{Estimated Travel Time} = \textit{Road Length}/\textit{Estimated Speed} \tag{18}$$

where

$$\textit{Estimated Speed} = \left(\sum_{i=1}^{n} \frac{d_i}{t_i}\right)/n \tag{19}$$

The estimated travel time is regularly updated, and the latest estimation is used by all vehicles to update their evacuation paths. Therefore, rerouting happens not only because of road closures but also abrupt speed changes, e.g. either the occurrence or relief of traffic jams.

Compared to distance-based re-routing (caused by road closures), the time-based re-routing is much harder to predict as it is related to dynamic traffic conditions, meanwhile the mechanism itself could also lead to a chain of reactions which further increase the complexity of the entire simulation. From a computational perspective, unexpected vehicle transitions could lead to higher communication cost, and possibly undesirable load balance. Moreover, both Voronoi and Target-shift proximity are designed for scenarios where vehicles (re)navigate entirely based on network distance, which is not the case in time-based rerouting scenarios.

Nonetheless, ViCTS maintains its effectiveness to a solid extent even in time-based rerouting situations. First of all, the fact that the center of evacuation zone is likely to be empty in the later stage of evacuation still holds regardless of the rerouting behaviors; which means that the Voronoi-based partition principle is still valid. Secondly, the distance is

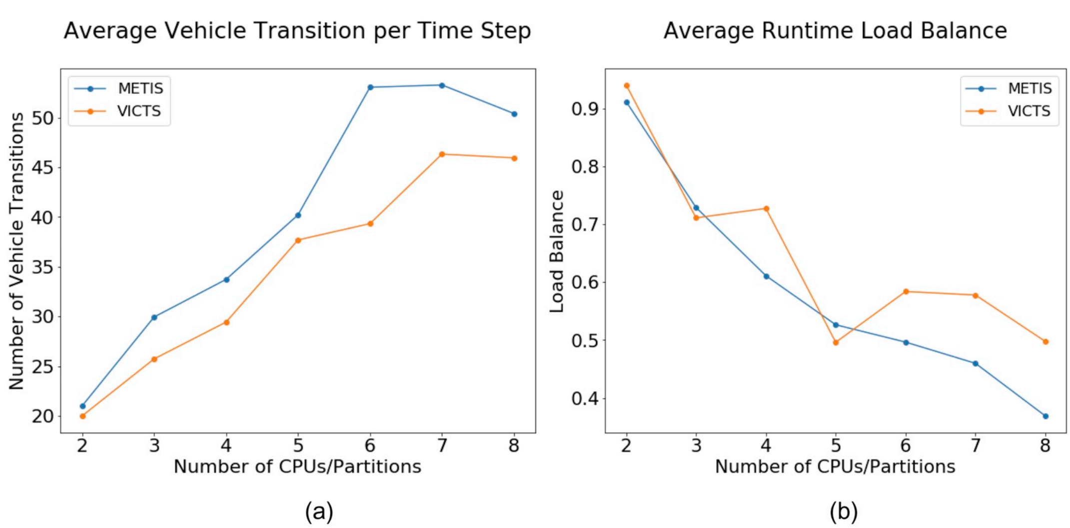

Fig. 9. Performance evaluation of ViCTS and METIS for time-based rerouting. (a) Communication cost (lower is better); (b) Load balance (higher is better). All experiments were conducted with a closure configuration 0.005/30.

a reasonable approximation for time, especially when a perfect travel time prediction is not available; therefore, the target-shift proximity developed from shortest distance is still able to reflect the target shift behavior in time-based situation. As shown in Fig. 9, although the gap has been reduced compared to the previous scenario, ViCTS still remains an advantage over METIS in both communication cost and load balance.

To further illustrate the computational difference between METIS and ViCTS in both scenarios. We conducted another set of comparison based on the computational time. As the total time span of the evacuation process is stochastically affected by the road closure sequences, comparing overall simulation time is not particularly meaningful. Instead, we focus on the ratio of time spent on synchronization (including communication and waiting) versus the time spent on computation across the entire evacuation process. At each time step, this ratio could be different among processors, as an comprehensive effect of load balance and communication cost. We select the largest ratio among all processors (the bottleneck) as the performance at that time step, and average that performance across the entire evacuation process. I.e.:

$$Overhead\ Ratio = Avg_t\left(Max_p \frac{T_{sync}}{T_{comp}}\right) \tag{20}$$

where t iterates over all time steps, and $p$iterates over all processors.

The measurement result is shown in Fig. 10. Generally, the overhead ratio (OR) increases as the number of partitions go up. Meanwhile, ViCTS outperforms METIS in both scenarios with lower ORs for most partition variations. This finding shows that ViCTS produces less overheads as the number of partitions increases, and thus achieves superior scalability.

Furthermore, we examine the effect of the road closure rate and interval to demonstrate the relative effectiveness of ViCTS in different road closure situations. Previous experiments are conducted using the parameter pair of 0.005/30, meaning each road has a probability of 0.5% to be closed every 30 s, which leads to about 55% road closure within an hour. In reality, a disaster could be better or worse. In order to test the proposed method in different disaster settings, we conduct a series of experiments using closure rates of 0.001, 0.005, 0.01, 0.05, and closure intervals of 15, 30, 60, and 120, for both distance- and time-based scenarios. For the illustrative purpose, we conduct these experiments using 8 partitions only; and the performance variance between ViCTS and METIS are presented as the ratio of METIS' OR to ViCTS' OR for the same parameter configuration, i.e.

$$Performance\ Variance = Overhead\ Ratio_{METIS}/Overhead\ Ratio_{ViCTS} \tag{21}$$

Therefore, a performance variance (PV) > 1 means that METIS generates more overhead than ViCTS; larger PVs indicates better performance of ViCTS over METIS, and vice versa.

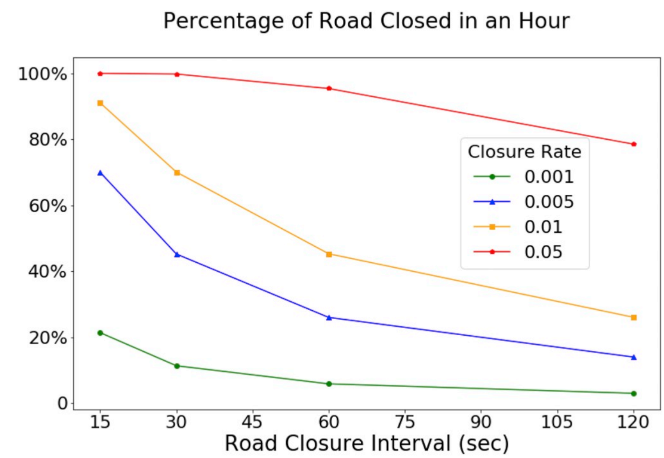

Fig. 11. One-hour road closure percentage under different rates and intervals.

The experiments are first grouped by closure rate, and a PV-interval curve is generated for each closure rate. The reason for such organization is that closure rate is more decisive for the severe magnitude of the disaster. For example, with a 0.05 closure rate, > 80% of roads will be closed in an hour even at 120 s closure interval, while the same percentage for a rate of 0.001 at 15 s closure interval is only about 20%. A more exhaustive illustration of road closure percentage at different rates and intervals are shown in Fig. 11. In fact, the 0.05 rate is so destructive that the evacuation simulation finishes shortly after beginning, as most vehicles are immediately stuck by massive road closures.

The final performance results are shown in Fig. 12. There are noticeable differences between distance- and time-based re-routing scenarios: 1) the PV values in distance-based re-routing scenarios (up to 3.0) is systematically larger than those in time-based scenarios (below 2.1); 2) the variance between different closure configurations is also significantly larger in distance-based (1.2–3.0) than time-based (1.2–2.1); 3) the performance pattern is clearer in distance-based scenarios, where smaller closure rate and larger intervals seem to have positive impacts on the performance variance despite some outliers. One common pattern revealed by both scenarios is that all PV values are > 1, meaning that ViCTS does outperform METIS regardless of scenarios and closure configurations.

The cross-scenario difference generally reinforces the previous conclusion that ViCTS excels more for distance-based re-routing, as

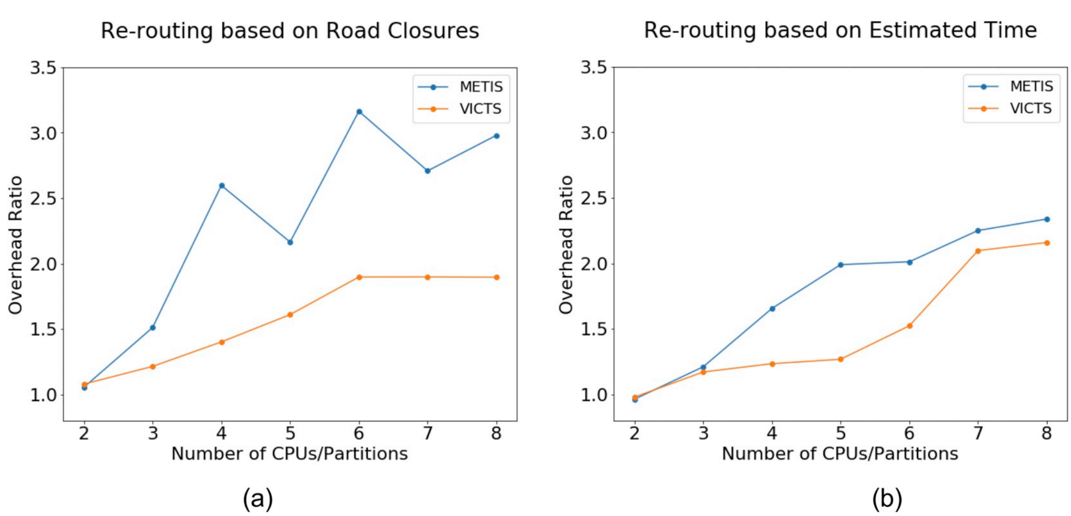

Fig. 10. Overhead ratio of METIS and ViCTS in both scenarios: (a) *Re*-routing based on road closures; (b) Re-routing based on estimated time.

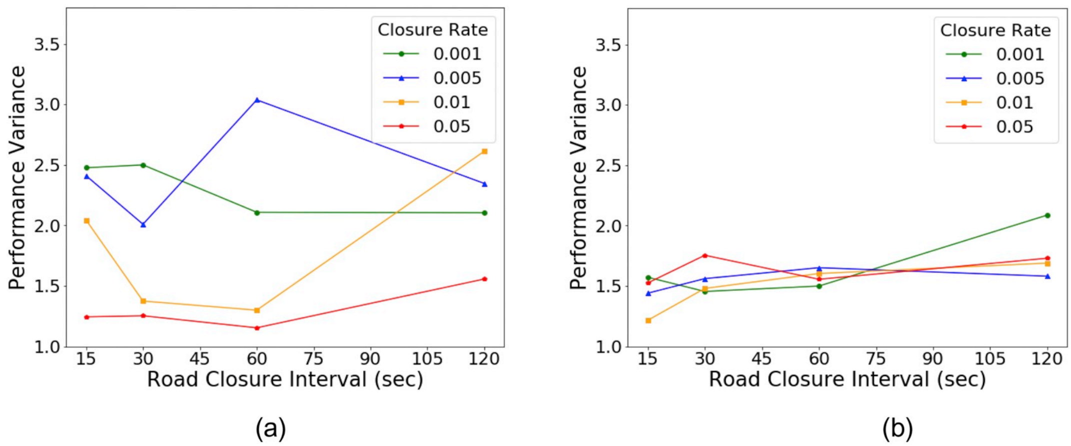

Fig. 12. Performance variance in different closure and re-routing scenarios: (a) re-routing based on road closures; (b) re-routing based on estimated time.

both Voronoi and Target-shift heuristics are designed based on distance. Nonetheless, it is still effective for the time-based scenario. As a bottom line, one could argue that the performance variance is around 1.5 for most cases (33% less overhead than METIS); and it goes higher in distance-based scenarios with moderate road closures, where the evacuation process can be completely unfolded, enabling ViCTS to fully demonstrate its advantage over the entire process.

5. Concluding discussion

In conclusion, a novel network partitioning algorithm - ViCTS - is developed to take spatial heterogeneity and dynamic uncertainty of emergency evacuation into account for achieving scalable agent-based modeling of emergency evacuation This research has demonstrated that ViTCS based on target-shift proximity outperforms the widely adopted traffic network partitioning algorithm, METIS, in terms of achieving computational load balance and minimizing communication cost.

Regarding overall computational performance, ViCTS consistently results in lower overhead ratios as compared with METIS, demonstrating better computational scalability. A direct implication is that adopting ViCTS for mass evacuation simulation is more computationally efficient and requires less execution time. This reduction is crucial to mass evacuation simulation, as emergency response is an extremely time-sensitive process. A more efficient computational model could result in more timely responses to save more lives and properties, especially in the context of deadly disasters that require mass evacuation.

Through our research on ViCTS network partitioning, it is evident that there are significant opportunities for future research towards optimally scalable agent-based traffic modeling for mass evacuation. For example, the partitioning solution does not necessarily need to be static throughout the entire simulation. A dynamic mechanism could be introduced to re-partition a spatial domain based on real-time vehicle distributions when load balance falls below a certain threshold. This improvement could be significant, as the exit routes of vehicles could change significantly due to congestion, which affects the shape of corresponding Voronoi diagrams (Ouyang & Daganzo, 2006; Ouyang, Wang, & Yang, 2015). Moreover, the spatiotemporal patterns of road closures caused by specific disasters could be incorporated in the partition design, as they could impact the road conditions as well as behaviors and decisions of individuals being evacuated (Chang, Peng, Ouyang, Elnashai, & Spencer Jr, 2012; Xie & Ouyang, 2019). Additionally, for network-based spatial representations such as street networks, it is helpful to optimize communication performance between processors according to the topological structure of spatial networks, e.g. by configuring fast data exchange channels for processors with heavy communication needs. This approach holds great potential to reduce the communication overhead and improve the overall computational performance. Following these research directions, the scalability of agent-based evacuation modeling could be further improved, enabling time critical decision making for mass evacuation.

Declaration of competing interest

The authors declare no conflicts of interests.

Acknowledgement

This work is supported in part by the U.S. National Science Foundation (NSF) under grant numbers: 1443080, 1551492, and 1664119. The authors thank Dr. Yan Liu at the CyberGIS Center for Advanced Digital and Spatial Studies for helpful discussions at the early stage of this research.

References

Baier, G., Erlebach, T., Hall, A., Köhler, E., Schilling, H., & Skutella, M. (2006). Length-bounded cuts and flows. *Automata, languages and programming* (pp. 679–690). .

Banks, J. H. (1990). Flow processes at a freeway bottleneck. *Transportation Research Record*(1287).

Barceló, J., et al. (1998). Microscopic traffic simulation for att systems analysis. A parallel computing version. *25th aniversary of CRT* (pp. 1–16). .

Boeing, G. (Sep. 2017). OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, Environment and Urban Systems, 65*, 126–139.

Brilon, W., Geistefeldt, J., & Regler, M. (2005). Reliability of freeway traffic flow: a stochastic concept of capacity. *Proceedings of the 16th International symposium on transportation and traffic theory. 125143*.

Chang, L., Peng, F., Ouyang, Y., Elnashai, A. S., & Spencer, B. F., Jr. (2012). Bridge seismic retrofit program planning to maximize postearthquake transportation network capacity. *Journal of Infrastructure Systems, 18*(2), 75–88.

Chen, X. (May 2008). Microsimulation of hurricane evacuation strategies of Galveston Island. *The Professional Geographer, 60*(2), 160–173.

Chen, X., Meaker, J. W., & Zhan, F. B. (Jul. 2006). Agent-based modeling and analysis of hurricane evacuation procedures for the Florida keys. *Natural Hazards, 38*(3), 321.

Cova, T. J., & Johnson, J. P. (Dec. 2002). Microsimulation of neighborhood evacuations in the urban–wildland interface. *Environment & Planning A, 34*(12), 2211–2229.

Dixit, V., & Wolshon, B. (Dec. 2014). Evacuation traffic dynamics. *Transportation Research Part C: Emerging Technologies, 49*, 114–125.

Farahmand, K. (1997). Application of simulation Modeling to emergency population evacuation. *Proceedings of the 29th conference on winter simulation, Atlanta, Georgia, USA* (pp. 1181–1188). .

Fiduccia, C. M., & Mattheyses, R. M. (1988). A linear-time heuristic for improving network partitions. *Papers on twenty-five years of electronic design automation* (pp. 241–247). .

Fieguth, P. (2016). *An introduction to complex systems: Society, ecology, and nonlinear dynamics.* Springer.

Fuchs, A. (2012). *Nonlinear dynamics in complex systems: Theory and applications for the life-, neuro- and natural sciences.* Springer Science & Business Media.

Gan, H.-S., Richter, K.-F., Shi, M., & Winter, S. (Sep. 2016). Integration of simulation and optimization for evacuation planning. *Simulation Modelling Practice and Theory, 67*, 59–73.

Haklay, M., & Weber, P. (Oct. 2008). OpenStreetMap: User-generated street maps. *IEEE*

Pervasive Computing, 7(4), 12–18.

Han, L. D., & Yuan, F. (2005). Evacuation modeling and operations using dynamic traffic assignment and most desirable destination approaches. *84th annual meeting of the transportation research board, Washington, DC* (pp. 2401–2406). .

Hendrickson, B., & Kolda, T. G. (Nov. 2000). Graph partitioning models for parallel computing. *Parallel Computing, 26*(12), 1519–1534.

Hey, T., & Trefethen, A. E. (May 2005). Cyberinfrastructure for e-science. *Science, 308*(5723), 817–821.

Karypis, G., & Kumar, V. (1995). *"METIS: Family of multilevel partitioning algorithms," wwwusers. cs. umn. edu/∖ karypis/metis/main. shtml*.

Karypis, G., Schloegel, K., & Kumar, V. (1997). *"Parmetis: Parallel graph partitioning and sparse matrix ordering library," Version 1. 0.* Dept. of Computer Science, University of Minnesota22.

Karypis, G., Aggarwal, R., Kumar, V., & Shekhar, S. (Mar. 1999). Multilevel hypergraph partitioning: Applications in VLSI domain. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 7*(1), 69–79.

Kernighan, B. W., & Lin, S. (1970). An efficient heuristic procedure for partitioning graphs. *The Bell System Technical Journal, 49*(2), 291–307.

Klefstad, R., Zhang, Y., Lai, M., Jayakrishnan, R., & Lavanya, R. (2005). A distributed, scalable, and synchronized framework for large-scale microscopic traffic simulation. *Proceedings. 2005 IEEE intelligent transportation systems, 2005* (pp. 813–818). .

Lawler, E. J., Thye, S. R., & Yoon, J. (2014). The emergence of collective emotions in social exchange. *Collective emotions*. Oxford: Oxford University Press.

Lu, Q., George, B., & Shekhar (2005). *Capacity constrained routing algorithms for evacuation planning: A summary of results.* Springer.

Miller-Hooks, E. (2001). Adaptive least-expected time paths in stochastic, time-varying transportation and data networks. *Networks, 37*(1), 35–52.

Murray-Tuite, P., & Wolshon, B. (Feb. 2013). Evacuation transportation modeling: An overview of research, development, and practice. *Transp. Res. Part C: Emerg. Technol. 27*, 25–45.

Okabe, A., Satoh, T., Furuta, T., Suzuki, A., & Okano, K. (Sep. 2008). Generalized network Voronoi diagrams: Concepts, computational methods, and applications. *International Journal of Geographical Information Science, 22*(9), 965–994.

Ouyang, Y., & Daganzo, C. F. (Feb. 2006). Discretization and validation of the continuum approximation scheme for terminal system design. *Transportation Science, 40*(1), 89–98.

Ouyang, Y., Wang, Z., & Yang, H. (Nov. 2015). Facility location design under continuous traffic equilibrium. *Transportation Research Part B: Methodological, 81*, 18–33.

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

Rickert, M., & Nagel, K. (Mar. 2001). Dynamic traffic assignment on parallel computers in TRANSIMS. *Future Generation Computer Systems, 17*(5), 637–648.

Sheffi, Y., Mahmassani, H., & Powell, W. B. (May 1982). A transportation network evacuation model. *Transportation Research Part A: General, 16*(3), 209–218.

Shekhar, S., et al. (Dec. 2012). Experiences with evacuation route planning algorithms. *International Journal of Geographical Information Science, 26*(12), 2253–2265.

Smith, L., Beckman, R., Anson, D., Nagel, K., & Williams, M. (Aug. 1995). *TRANSIMS: TRansportation ANalysis and SIMulation system.* NM (United States): Los Alamos National Lab LA-UR-95-1664; CONF-9504197-1.

Sun, X., Chen, F., Li, X., & Wang, X. (2012). An improved dynamic load balancing algorithm for parallel microscopic traffic simulation. *Proceedings of 2012 international conference on measurement, information and control. 2. Proceedings of 2012 international conference on measurement, information and control* (pp. 600–604).

Suzumura, T., Houngkaew, C., & Kanezashi, H. (2014). Towards billion-scale social simulations. *Proceedings of the winter simulation conference 2014* (pp. 781–792). .

Thulasidasan, S., Kasiviswanathan, S. P., Eidenbenz, S., & Romero, P. (2010). Explicit spatial scattering for load balancing in conservatively synchronized parallel discrete event simulations. *Proceedings of the 2010 IEEE workshop on principles of advanced and distributed simulation* (pp. 150–158). .

Tu, H., van Rij, M., Henkens, N., & Heikoop, H. (2010). Empirical investigation on stochastic Dutch freeway capacity. *13th international IEEE conference on intelligent transportation systems* (pp. 825–830). .

U.S. Census Bureau *Population estimation of Miami city, Florida*. [Online]. Available https://www.census.gov/quickfacts/fact/table/miamicityflorida, miamidadecountyflorida/PST045217, Accessed date: 1 November 2018.

Voronoi, G. (1908). Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Deuxième mémoire. Recherches sur les parallélloèdres primitifs. *Journal fur die Reine und Angewandte Mathematik, 134*, 198–287.

Wang, S. (2008). Formalizing computational intensity of spatial analysis. *Proceedings of the 5th international conference on geographic information science* (pp. 184–187). .

Wei, D., Chen, F., & Sun, X. (2010). An improved road network partition algorithm for parallel microscopic traffic simulation. *2010 international conference on mechanic automation and control engineering* (pp. 2777–2782). .

Wen, Y. (2008). Scalability of dynamic traffic assignment (Doctoral dissertation). *Massachusetts Institute of Technology*.

Wolshon, B. (Aug. 2008). Empirical characterization of mass evacuation traffic flow. *Transportation Research Record: Journal of the Transportation Research Board, 2041*, 38–48.

Wolshon, B., & McArdle, B. (Mar. 2009). Temporospatial analysis of hurricane Katrina regional evacuation traffic patterns. *Journal of Infrastructure Systems, 15*(1), 12–20.

Xie, S., & Ouyang, Y. (Feb. 2019). Reliable service systems design under the risk of network access failures. *Transportation Research Part E: Logistics and Transportation Review, 122*, 1–13.

Xu, Y., & Tan, G. (2012). hMETIS-based offline road network partitioning. *AsiaSim 2012* (pp. 221–229). Berlin, Heidelberg: Springer.

Xu, Y., Cai, W., Aydt, H., & Lees, M. (2014). Efficient graph-based dynamic load-balancing for parallel large-scale agent-based traffic simulation. *Proceedings of the winter simulation conference. 2014. Proceedings of the winter simulation conference* (pp. 3483–3494).

Xu, Y., Cai, W., Eckhoff, D., Nair, S., & Knoll, A. (2017). A graph partitioning algorithm for parallel agent-based road traffic simulation. *Proceedings of the 2017 ACM SIGSIM conference on principles of advanced discrete simulation* (pp. 209–219). .

Yan, X. (2014). *City-scale traffic simulation-performance and calibration (Doctoral dissertation).* National University of Singapore.

Yin, W., Murray-Tuite, P., Ukkusuri, S. V., & Gladwin, H. (May 2014). An agent-based modeling system for travel demand simulation for hurricane evacuation. *Transp. Res. Part C: Emerg. Technol. 42*, 44–59.

Yin, D., Liu, Y., Padmanabhan, A., Terstriep, J., Rush, J., & Wang, S. (2017). A CyberGIS-Jupyter framework for geospatial analytics at scale. *Proceedings of the practice and experience in advanced research computing 2017 on sustainability, success and impact, New Orleans, LA, USA* (pp. 18:1–18:8). .

Yuan, S., Chun, S. A., Spinelli, B., Liu, Y., Zhang, H., & Adam, N. R. (May 2017). Traffic evacuation simulation based on multi-level driving decision model. *Transp. Res. Part C: Emerg. Technol. 78*, 129–149.