Simulation-based Evacuation Planning for Urban Areas

Theodoros Chondrogiannis
University of Konstanz
Konstanz, Germany
theodoros.chondrogiannis@uni.kn

Panagiotis Bouros
Johannes Gutenberg University of Mainz
Mainz, Germany
bouros@uni-mainz.de

Winfried Emser
University of Konstanz
Konstanz, Germany
winfried.emser@uni.kn

ABSTRACT

Evacuation planning is a critical task in disaster management. Especially in situations such as natural disasters or terrorist attacks, large crowds need to move away from danger and reach designated safe zones. For this purpose, various approaches that efficiently compute evacuation plans in urban areas have been proposed. To evaluate the computed plans, previous works employ heuristics that can only roughly estimate the egress time of each plan. Intuitively, a much better approach is to estimate the egress time via simulation. However, designing a simulation model is usually a time-consuming task and, what is more, this model can only be used to evaluate evacuation plans for a specific area. In this paper, we address these issues presenting EURASIM. Our system enables the automated generation of simulation models for urban areas. Furthermore, EURASIM is designed in a way that algorithms for evacuation planning can be easily integrated, thus functioning as a testbed for the development of even better solutions.

KEYWORDS

Evacuation Planning, Simulation, Disaster Management

ACM Reference Format:
Theodoros Chondrogiannis, Panagiotis Bouros, and Winfried Emser. 2021. Simulation-based Evacuation Planning for Urban Areas. In *29th International Conference on Advances in Geographic Information Systems (SIGSPATIAL '21), November 2–5, 2021, Beijing, China.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3474717.3483963

1 INTRODUCTION

Evacuation planning is a critical task in disaster management. Especially in situations such as natural disasters or terrorist attacks, large crowds need to move away from danger and reach designated safe zones. To avoid panic and potential catastrophic collisions while dealing with the aftermath of such events, the availability of safe and efficient evacuation plans is necessary. For instance, during the Hurricane Irma events in Florida, people blindly evacuated without considering potential risks (e.g., traffic congestion, movement-conflicts, etc.), which caused major delays. A technical report by Wong et al. [12] on the behavior of evacuees states that issuance of complete and clear mandatory evacuation orders is of utmost importance. To this end, the main goal of evacuation planning for urban areas is to compute safe routes that minimize the egress time, i.e, the total time a crowd needs to move from a danger zone to a safe area.

To compute evacuations plans, various algorithmic approaches have been proposed, most of which treat evacuation planning as a flow optimization problem. Yamada [13] proposed a model with capacity constraints for safe areas and a method that computes evacuation plans by solving a minimal cost flow problem. The *Capacity Constrained Route Planner* (CCRP), originally introduced by Lu et al. [7] but later improved by Yin [14], and by Gupta and Sanda [3], also treats evacuation planning as a flow optimization problem. However, in addition to capacities for the safe areas, CCRP also considers capacities for all nodes and edges of the road network, thus introducing additional constraints. On top of these constraints, Herschelman et al. [4, 5] proposed to also consider possible collisions that are expected to increase the egress time. To this end, the authors presented algorithms which aim at computing an evacuation plan that minimizes such collisions.

The aforementioned works enable the computation of evacuation plans in urban areas. Some of them even provide a detailed schedule that the evacuees should follow in order to evacuate a danger zone in an orderly fashion. However, with regard to the quality of the computed plans, these approaches offer only an estimate of the egress time using heuristics. A more sophisticated way to evaluate computed evacuation plans is to simulate them, thus computing the egress time more accurately [2, 8, 9]. Furthermore, apart from obtaining a more accurate egress time, Islam et al. [6] showed recently that the results of a simulation can be used as feedback that allows the improvement of an already computed evacuation plan. Unfortunately, in order to execute a simulation, it is necessary to carefully design a simulation model, a process usually done by hand and only for a specific area. Evaluating the effectiveness of algorithmic solutions and simulating the computed evacuation plans is a task that requires significant time and effort.

To address the aforementioned issues, in this paper we propose EURASIM[1], a system that enables the *automated generation* of simulation models to evaluate evacuations plans in urban areas. First, EURASIM applies an algorithm to compute an evacuation plan from one or more danger zones to one or more safe areas. EURASIM takes into account the characteristics of the edges in the road network, e.g. length and estimated width, to compute capacities for each edge and the average speed of the evacuees. Afterwards, a simulation model is automatically generated which is then fed into a simulator to obtain, among other information, the egress time for the computed evacuation plan. Finally, the evacuation plan is visually presented to the user. Note that our system is designed with extensibility in mind, i.e., it facilitates the integration of evacuation planning algorithms. Under this premise, EURASIM can also be used as a testbed for comparing evacuation planning solutions.

[1] https://eurasim.github.io

SIGSPATIAL '21, November 2–5, 2021, Beijing, China

ACM ISBN 978-1-4503-8664-7/21/11.
https://doi.org/10.1145/3474717.3483963

2 SYSTEM OVERVIEW

We now present EURASIM, a system that streamlines the evaluation of evacuation plans for urban areas via simulation. More specifically, our system allows users to compute an evacuation plan, generates a simulation model, and simulates the evacuation plan in order to compute its egress time, i.e., the time required for the last evacuee to reach their designated safe area. EURASIM is implemented in Java using the JGraphT[2] library for the network model and the implementation of the algorithms, uses PostGIS[3] to store spatial and road network data, and employs Mapbox[4] to display the map and visualize results. In what follows, we describe the modules of EURASIM shown in Figure 1 and the function each module serves.

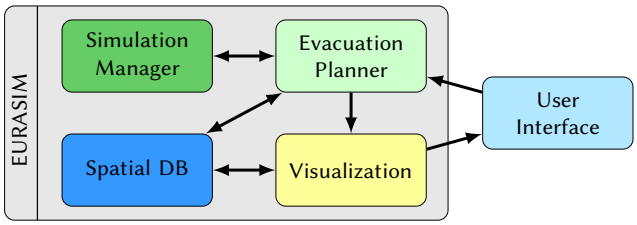


Figure 1: System architecture.

2.1 Spatial DB

The *Spatial DB* module is responsible for storing the data for the road network on which evacuation plans are computed. More specifically, we first obtain road network data from OpenStreetMap[5] (OSM) and we then transform the data into a routable graph format. The resulting graph is stored as an edge list in a single table in PostGIS. Every tuple stores all information related to a particular edge and its two adjacent nodes, i.e., edge length, source and target node location, edge type, and edge shape as a sequence of coordinates. In addition to storing road network data, the module is also responsible for executing necessary spatial operations, e.g., mapping input coordinates to nodes/edges of the road network. All spatial queries are executed in PostGIS.

2.2 Evacuation Planner

The *Evacuation Planner* module is responsible for computing evacuation plans while considering capacity and flow constraints. In what follows, we describe the two components of this module, i.e., the *Network Model* and the *Evacuation Plan Computation*.

2.2.1 Network Model. To enable the computation of evacuation plans, the Evacuation Planner represents a road network as a *directed weighted graph* $G = (N, E)$. Each edge $e \in E$ is assigned: a) a positive value $\ell(e)$ that is the length of the edge, and b) a second positive value $w(e)$ that is the width of the street the edge represents. While the length of an edge is given by the OSM data, the width has to be estimated. To do so, we consider the type of street the edge is associated with as provided by OSM. We split all edges into groups based on these two parameters. Then, we select a few edges from each group and we estimate the width of those edges manually. The width of each edge is defined as the average width of the sampled edges in each group.

Furthermore, for each edge e we need to estimate its capacity $c(e)$, i.e., the maximum number of people that can be on the edge at any given time. The capacity of an edge e is given by

$$c(e) = \rho_{max} \cdot w(e) \cdot \ell(e)$$

where ρ_{max} is the maximum crowd density, i.e., the maximum number of people per m^2. Existing works have proposed different values for crowd density [10]. As there is no settled way to set the ρ_{max} value, we define it as a system parameter and we allow users to set the value themselves.

Figure 2a illustrates a sample graph representing a road network. Each edge is assigned a triple $\{\ell, w, c\}$. The capacity c for all edges is computed assuming, without loss of generality, that $\rho_{max} = 5$ people$/m^2$. For illustration purposes, we do not draw the direction of the edges; all edges are assumed to be bidirectional.

2.2.2 Evacuation Plan Computation. In order to compute an evacuation plan, the module takes as input a query $q(S, T)$, where S is the set of danger zones that need to be evacuated, and T is the set of safe areas to which the evacuees must be routed. Both each danger zone and each safe area $n \in S \cup T$ are represented by a pair $(\{x, y\}, k)$, where $\{x, y\}$ is the longitude-latitude coordinates of n[6]. For the danger zones, k denotes the number of evacuees that need to leave the zone, while for safe areas k denotes the maximum number of evacuees that can be routed to this area, i.e., the maximum capacity of the safe area.

The first step for processing an input query $q(S, T)$ is the mapping of the real-world coordinates to the road network. For each danger zone or safe area $a \in S \cup T$ with location p_a, the Evacuation Planner first communicates with Spatial DB and retrieves the nearest point p_n of the road network, the edge (u, v) for which p_n lies on, and the locations p_u and p_v of the endpoints of (u, v). A new node n with the same coordinates as p_n is temporarily added to the network representing a. Then, edge e is temporarily removed from the graph and is replaced by the temporary edges (u, n) and (n, v). The two new edges have the same width and capacity as e, while the lengths $\ell(u, n)$ and $\ell(n, v)$ are

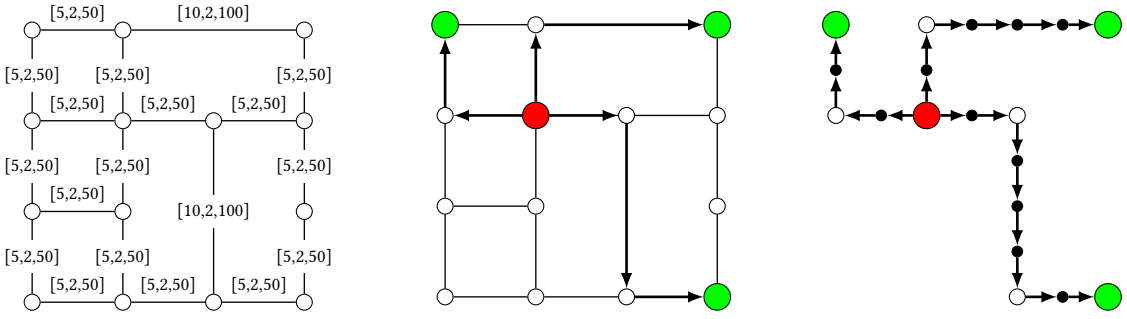
$$\ell(u, n) = \frac{d_g(u, n) \cdot \ell(u, v)}{d_g(u, v)} \text{ and } \ell(n, v) = \frac{d_g(n, v) \cdot \ell(u, v)}{d_g(u, v)},$$

where $d_g(u, v)$ is the geodesic distance between the locations of nodes u and v. Note that, to accelerate future planning tasks, for locations that are stored in Spatial DB permanently, i.e., known points of interest, the aforementioned mapping changes to the network are permanent.

Having mapped all input locations to nodes of the graph, the input query is transformed from a query between source and target location to source and target nodes. At this point, the Evacuation Planner employs an evacuation planning algorithm to compute the plan. The resulting evacuation plan is a set of paths P_E such that each path connects one source (danger zone) to one target (safe

[2] https://jgrapht.org
[3] http://postgis.net
[4] https://www.mapbox.com
[5] https://www.openstreetmap.org/

[6] While danger zones and safe areas are represented by points, both can also be represented by polygons and associated with multiple nodes of the road network.



(a) Original network (b) Evacuation plan (c) Simulation graph

Figure 2: Illustration of the original network, a computed evacuation plan, and the graph on which the simulation is executed.

area). Note that our system imposes no limit on the number of paths between each source-target pair, i.e., there might be multiple routes for evacuees to move from a danger zone to the same safe area.

After the computation of the evacuation plan is finished, P_E is forwarded to the Simulation Manager, in order to execute the simulation and compute the egress time. As soon as the Simulation Manager returns the egress time along with additional statistics, the plan is forwarded to the Visualization module in order to be presented to the user. Figure 2b illustrates an evacuation plan in our sample network. The red node represents a danger zone, the green nodes represent three safe areas, and the arrows indicate the routes the evacuees must follow.

2.3 Simulation Manager

The *Simulation Manager* module is responsible for the simulation of a given evacuation plan and the computation of its egress time along with some additional statistics, e.g., density of edges over time. For the simulation, the module employs NetLogo, an agent-based simulation modeling platform [11]. The Simulation Manager takes as input an evacuation plan P_E from the Evacuation Planner and generates a simulation model that is then fed to NetLogo. The evacuees are modeled as agents that follow an assigned path from the source node (danger zone) they are located in, to the target node (safe zone) they need to go, according to P_E.

In NetLogo, the passing of time is modeled using ticks, i.e., one tick signifies the passing of one unit of time. To model the movement of evacuees over time accurately the module generates a *simulation graph*. First, all nodes in the evacuation plan are added to the simulation graph. Then, each edge in the evacuation plan is split into multiple segments based on the speed of the evacuees by introducing *intermediate nodes*. For a given edge e the number of intermediate nodes $N_i(e)$ is

$$|N_i(e)| = \frac{\ell(e)}{\text{speed} \cdot \text{ticks/second}} - 1.$$

The *speed* is the maximum speed in which evacuees should move in order to minimize the egress time while minimizing the danger for accidents. As various values have been proposed in the bibliography [1, 8], our model treats speed as a user-defined parameter. In our model, a tick increase happens after every evacuee has been asked to move from one node to the next on their assigned path in the simulation graph.

Figure 2c illustrates the simulation graph generated from the evacuation plan of Figure 2b for our sample network. For exemplifying the simulation of the evacuation plan, consider there are 300 evacuees in the single danger zone (red node), each of the three safe zones (green nodes) has a capacity of 100. We assume that according to the evacuation plan, the 300 evacuees are routed evenly to the safe areas. The capacity of all edges is 25, computed after dividing the capacity of the original edge by the number of intermediate nodes + 1. Hence, evacuees routed towards a safe area have to move in groups of 25, yielding a total of 12 groups. The units of time each group requires in order to reach their designated safe area equal the number of nodes on their assigned path in the simulation graph. Since we assumed the same number of evacuees for all safe areas, the egress time of the evacuation plan is given by the longest route. That is the route to the bottom right safe area, which yields an egress time of 9 units of time.

2.4 Visualization

The visualization module is responsible for presenting the results after an evacuation plan is computed. First the module retrieves the shapes of all edges contained in the shortest path from Spatial DB and produces a response in GeoJSON format. Apart from the routes themselves, the module also enables the visualization of the density of edges over time.

3 USER INTERFACE & DEMONSTRATION

In this section we describe the user interface of EURASIM, and we show how our system can be used to compute, simulate, and visualize evacuation plans. Figure 3 shows the interface of our system. On the left hand side, the user can explore points of interest and set the input to the evacuation planning algorithm. The system also allows users to set system parameters such as crowd density and pedestrian speed. For our demonstration, we use the road network and points of interest in the city of Berlin. During the demonstration, users can explore the features of the system and use it to compute evacuation plans that would be useful in different emergency scenarios, e.g., earthquakes, buildings on fire etc.

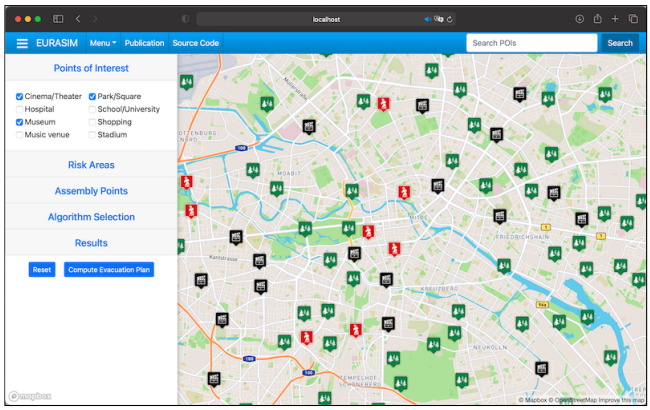


Figure 3: EURASIM User Inteface

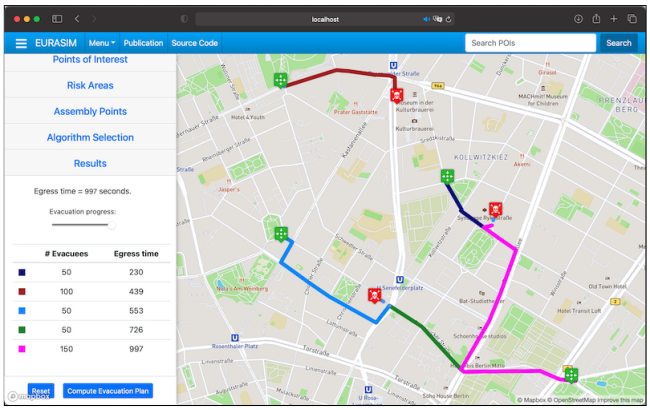


Figure 4: Visualized evacuation plan

The computation of an evacuation plan requires three steps. First, the user selects the danger zones that need to be evacuated and sets the number of evacuees per zone. Second, the user selects the safe zones where the evacuees must be routed to and sets the capacity of each area. A user can select danger zones and safe areas either by clicking on the map, or by selecting them among a set of points of interest that are stored in the system. For example, Figure 3 shows museums and cinemas in the city of Berlin some of which might have to be evacuated during an emergency, and several parks that can be selected as safe areas. As the final step, the user selects the algorithm used to compute the plan. For this step, our system integrates the well-established *Capacity Constraint Route Planner* (CCRP) algorithm [7]. However, despite employing only CCRP for this demonstration, our system is implemented with extensibility in mind. By simply implementing a Java interface and with minor extensions in the user interface, more algorithms can be easily integrated into our system allowing the user to compute and compare multiple evacuation plans.

After the user sets the parameters, the system proceeds with the computation of the evacuation plan. As soon as the computation is finished, the evacuation plan is visualized on the map. Under the Results tab, information about the evacuation plan is displayed. More specifically, the results include the *estimated* egress time given by the evacuation plan algorithm using heuristics, the egress time computed by the simulator, and the list of routes from danger zones to safe areas. For each route, the number of evacuees that will follow this route according to the evacuation plan, and the time necessary for the evacuees to reach the destination safe zone is also displayed. Finally, the interface also provides a range control, which allows the user to explore the progress of the number of evacuees on each danger zone and the available capacity of each safe zone over time. Figure 4 illustrates an evacuation plan that involves three danger zones and four safe areas in the city of Berlin.

4 CONCLUSIONS & FUTURE WORK

We have presented EURASIM, a system that enables the automatic generation of simulation models for evaluating evacuation plans for urban areas. The system is not intended to be used solely as a supporting tool for urban planning, but also as a testbed for the development of novel algorithmic solutions. In the future, we plan to conduct a thorough evaluation of existing state-of-the-art methods for computing evacuation plans using EURASIM. We also plan to integrate in our system models for human mobility to enable the execution of more elaborate simulations. Furthermore, we will investigate solutions to enable large-scale simulations.

ACKNOWLEDGMENTS

This work is partially supported by Grant No. CH 2464/1-1 of the Deutsche Forschungsgemeinschaft (DFG).

REFERENCES

[1] Gabriele Bernardini, Marco D'Orazio, Enrico Quagliarini, and Luca Spalazzi. 2014. An Agent-based Model for Earthquake Pedestrians' Evacuation Simulation in Urban Scenarios. *Transportation Research Procedia* 2 (2014), 255–263.
[2] Heng-Soon Gan, Kai-Florian Richter, Mingzheng Shi, and Stephan Winter. 2016. Integration of Simulation and Optimization for Evacuation Planning. *Simulation Modelling Practice and Theory* 67 (2016), 59–73.
[3] Ajay Gupta and Nandlal L. Sarda. 2014. Efficient Evacuation Planning for Large Cities. In *Proc. of the DEXA Conf.* 211–225.
[4] Roxana Herschelman, Ahmad Qutbuddin, and KwangSoo Yang. 2021. Conflict-Free Evacuation Route Planning. *GeoInformatica* (2021), 1–24.
[5] Roxana Herschelman and KwangSoo Yang. 2019. Conflict-Free Evacuation Route Planner. In *Proc. of the ACM SIGSPATIAL Conf.* 480–483.
[6] Kazi Ashik Islam, Madhav Marathe, Henning Mortveit, Samarth Swarup, and Anil Vullikanti. 2020. A Simulation-based Approach for Large-scale Evacuation Planning. In *Proc. of the IEEE BigData Conf.* 1338–1345.
[7] Qingsong Lu, Betsy George, and Shashi Shekhar. 2005. Capacity Constrained Routing Algorithms for Evacuation Planning: A Summary of Results. In *Proc. of the SSTD.* 291–307.
[8] Erick Mas, Anawat Suppasri, Fumihiko Imamura, and Shunichi Koshimura. 2012. Agent-based Simulation of the 2011 Great East Japan Earthquake/Tsunami Evacuation: An Integrated Model of Tsunami Inundation and Evacuation. *Journal of Natural Disaster Science* 34, 1 (2012), 41–57.
[9] Hyeong Suk Na and Amarnath Banerjee. 2019. Agent-based Discrete-event Simulation Model for No-notice Natural Disaster Evacuation Planning. *Computers & Industrial Engineering* 129 (2019), 44–55.
[10] Hendrik Vermuyten, Jeroen Beliën, Liesje De Boeck, Genserik Reniers, and Tony Wauters. 2016. A Review of Optimisation Models for Pedestrian Evacuation and Design Problems. *Safety Science* 87 (2016), 167–178.
[11] Uri Wilensky. 1999. NetLogo. Center for Connected Learning and Computer-based Modeling, Northwestern University. http://ccl.northwestern.edu/netlogo/
[12] Stephen Wong, Susan Shaheen, and Joan Walker. 2018. *Understanding Evacuee Behavior: A Case Study of Hurricane Irma.* Technical Report.
[13] Takeo Yamada. 1996. A network flow approach to a city emergency evacuation planning. *International Journal of Systems Science* 27, 10 (1996), 931–936.
[14] Dafei Yin. 2009. A Scalable Heuristic for Evacuation Planning in Large Road Network. In *Proc. of IWCTS Workshop.* 19–24.