Towards system optimum: Finding optimal routing strategies in time-dependent networks for large-scale evacuation problems *

Gregor Lämmel[1] and Gunnar Flötteröd[2]

[1] Berlin Institute of Technology (TU Berlin), Transport Systems Planning and Transport Telematics, Berlin, Germany
laemmel@vsp.tu-berlin.de
[2] Ecole Polytechnique Fédérale de Lausanne (EPFL), Transport and Mobility Laboratory, Lausanne, Switzerland
gunnar.floetteroed@epfl.ch

Abstract. Evacuation planning crucially depends on good routing strategies. This article compares two different routing strategies in a multi-agent simulation of a large real-world evacuation scenario. The first approach approximates a Nash equilibrium, where every evacuee adopts an individually optimal routing strategy regardless of what this solution imposes on others. The second approach approximately minimizes the total travel time in the system, which requires to enforce cooperative behavior of the evacuees. Both approaches are analyzed in terms of the global evacuation dynamics and on a detailed geographic level.

1 Introduction

The evacuation of whole cities or even regions is a problem of substantial practical relevance, which is demonstrated by recent events such as the evacuation of Houston because of Hurricane Rita or the evacuation of coastal cities in the case of tsunamis.

The development of evacuation simulations relies strongly on results obtained in the field of transportation modeling. Like in transportation, one can distinguish static approaches, e.g., [18], and dynamic approaches, e.g., [16]. A typical static evacuation simulation is MASSVAC [7]. The obvious shortcoming of static models is that they do not capture dynamic effects, which are highly relevant in evacuation situations. Consequently, many dynamic traffic assignment (DTA) models have been applied for evacuation simulations, e.g., MITSIM [8], DYNASMART [10], and PARAMICS [3].

Another aspect according to which transportation models may be classified is their granularity: Microscopic models represent every trip-maker individually, whereas macroscopic models aggregate traffic into continuous streams. All of

* The original publication is available at www.springerlink.com

the above DTA packages rely on microscopic traffic models. Further microscopic approaches that have been applied to the simulation of evacuation dynamics are cellular automata [9] and the social force model [6]. Random utility models are also applicable to the microscopic modeling of pedestrian dynamics, however, they are yet to be applied in evacuation scenarios [2]. Examples of software packages based on macroscopic models are ASERI [17] and Simulex (www.iesve.com).

This paper evaluates the following two routing strategies with a learning-based multi-agent (micro)simulation in a real-world evacuation scenario: 1. A strategy where every agent learns an evacuation route of minimal travel time, regardless of the consequences for others. This selfish learning behavior leads towards a Nash equilibrium, where nobody can gain by switching to a different route. This strategy is called "user optimal" in transportation. 2. A "system optimal" strategy, where the total travel time of all agents is minimized. Here, learning agents are no longer optimizing their individual travel times only but in some way also care about others.

The added value of the agent-based approach is its natural representation of individual travelers as software agents that interact in a simulated version of the real world (a virtual environment). The agent-based approach has an edge over macroscopic models in that it allows (at least technically) for a much higher model resolution. However, this comes at the price of greater difficulties in its mathematical treatment. The agent-based routings presented in this article are therefore only of an approximate nature, and they are enforced exclusively by modifying the information provided to replanning agents.

The remainder of this article is organized as follows. Section 2 outlines the simulation framework. Section 3 describes the investigated routing strategies in detail. Section 4 presents simulation results, and Section 5 concludes the article.

2 Simulation framework

We implement our experiments in the MATSim simulation framework. Since the details of this system are described elsewhere, e.g., [11] and www.matsim.org, only a brief description is given here.

MATSim always starts with a synthetic population, which is based as much as possible on existing information such as census data. Every synthetic individual possesses one or several plans. These plans represent the different traveling intentions of that individual. In an evacuation context, a plan corresponds to a route from an individual's current location to a safe place. Plans are generated by an iterative learning mechanism. In every iteration, one plan is selected by every agent for execution in the virtual environment. The learning logic tests different plans, eventually discards inferior plans, and sometimes generates new plans [4].

The virtual environment is a pedestrian traffic flow simulation, where each street (link) is represented by a first-in/first-out queue with three parameters [5]: minimum link traversal time, maximum link outflow rate (in evacuees per time unit), and link space capacity (in evacuees). The link space capacity limits

Algorithm 1 Nash equilibrium routing

1. initialize $\tau_a(k)$ with the free-flow travel time for all links a and time steps k
2. repeat for many iterations:
   (a) recalculate routes based on link costs $\tau_a(k)$
   (b) load vehicles on network, obtain new $\tau_a(k)$ for all a and k

the number of agents on the link and generates spillback if the link is filled up. In the context of a tsunami evacuation, an additional difficulty results from the fact that a flooded link becomes unavailable. Reference [11] describes in detail how this issue is resolved.

3 Routing solutions

Each agent iteratively adjusts its evacuation plan during the simulation. After each iteration, every agent calculates the cost of the most recently executed plan. Based on this cost, the agent revises its plans. Some agents generate new plans using a time-dependent Dijkstra algorithm. The other agents select an existing plan, which they have previously used. This selection is realized through a Multinomial Logit model, e.g., [1], that stabilizes the simulation dynamics by allowing somewhat inferior plans to be considered for execution as well.

In the following, we discuss two different cost functions that approximately lead either to user optimal or to system optimal routing solutions. Note that we modify the agents' routing behavior only by adjusting the costs based on which the routing and the plan choice are conducted, but we do not change the replanning logic itself. For simplicity, we subsequently omit the attribute "approximate" in conjunction with either strategy.

3.1 Nash equilibrium approach

In a Nash equilibrium, no agent can gain by unilateral deviation from its current plan [14]. The cost function provided to replanning agents in the Nash equilibrium approach only comprises travel times. Formally, the real-valued time is discretized into K segments ("bins") of length T, which are indexed by $k = 0...K-1$. The time-dependent link travel time when entering link a in time step k is denoted by $\tau_a(k)$. Alg. 1 drafts the Nash-equilibrium routing logic.

3.2 System optimal approach

A system optimal routing solution minimizes the total travel time in the system. Although a system optimum is a cooperative routing strategy, it can be obtained by the same self-serving routing logic that is employed to calculate a Nash equilibrium. The only difference is that for a system optimum, the travel time based on which agents evaluate their routes needs to be replaced by the marginal travel time [15]. The marginal travel time of a route is the amount by which the total system travel time changes if one additional vehicle drives along that route. It is the sum of the cost experienced by the added vehicle and the cost imposed on other vehicles. The latter is denoted here as the "social cost". The subsequently developed approximation of the social cost term is based on continuous quantities and ignores for simplicity the interrelations of different links in the network. A discretized version is given at the very end.

Assume that the "causative" agent (unit) for which we would like to calculate the social cost it generates is of mass (size) dn and enters the considered isolated link at time t_0. If there is no congestion on the link, the agent can leave the link unhindered after the free-flow travel time τ^{free} and does not incur any cost on other agents further upstream. If there is congestion, however, there also is a positive social cost, which can be calculated in the following way.

The effect of the causative agent persists only as long as the queue it went through persists – the only trace it can possibly leave in the system is a changed state of this queue. Assume that the queue encountered when entering the link at t_0 dissolves at $t^{\text{e}}(t_0)$. Now, consider another "affected" agent that enters the link at $t_1 > t_0$, and assume that this agent leaves the link before $t^{\text{e}}(t_0)$. Denote by $n(t_1)$ the occupancy (in agent units) of the link at the affected agent's entry time t_1 and by $Q^{\text{out}}(t)$ the accumulated outflow (in agent units) of the link until time t. The exit time t_2 of the affected agent solves

$$\begin{aligned} Q^{\text{out}}(t_2) - Q^{\text{out}}(t_1) &= n(t_1) \\ &\Rightarrow t_2 = (Q^{\text{out}})^{-1}(n(t_1) + Q^{\text{out}}(t_1)). \end{aligned} \quad (1)$$

Denote by $d\tau(t_1)$ the additional travel time experienced by the affected agent because of the causative agent. If the latter had not entered the link, the following would hold:

$$\begin{aligned} Q^{\text{out}}(t_2 - d\tau(t_1)) - Q^{\text{out}}(t_1) &= n(t_1) - dn \\ &\Rightarrow t_2 = d\tau(t_1) + (Q^{\text{out}})^{-1}(n(t_1) - dn + Q^{\text{out}}(t_1)). \end{aligned} \quad (2)$$

A combination of (1) and (2) yields

$$d\tau(t_1) = (Q^{\text{out}})^{-1}(n(t_1) + Q^{\text{out}}(t_1)) - (Q^{\text{out}})^{-1}(n(t_1) - dn + Q^{\text{out}}(t_1)). \quad (3)$$

In order to calculate the social cost $C(t_0)$ generated by the causative agent, these terms are integrated over the entire span of entry times during which the queue at the downstream end of the link is encountered:

$$C(t_o) = \int_{t_1=t_o}^{t^{\text{e}}(t_0)-\tau^{\text{free}}} d\tau(t_1) q^{\text{in}}(t_1) dt_1 \quad (4)$$

where $q^{\text{in}}(t_1)$ is the entry flow rate at t_1 such that $q^{\text{in}}(t_1)dt_1$ is the affected agent mass entering at t_1.

In the following, a simplification of (4) is presented. Stationary flow conditions are assumed in that $q^{\text{in}}(t) \equiv q^{\text{out}}(t) \equiv \bar{q}$, which implies $Q^{\text{out}}(t) \equiv \bar{q}t$ and

Algorithm 2 System optimum approach

1. initialize $C_a(k) \equiv 0$ and $\tau_a(k) \equiv \tau_a^{\text{free}}$ for all links a and time steps k
2. repeat for many iterations:
   (a) recalculate routes based on link costs $\tau_a(k) + C_a(k)$
   (b) load vehicles on network, obtain new $\tau_a(k)$ for all a and k
   (c) for all links a, identify congestion durations:
      i. $k^{\text{e}} = K$
      ii. for $k = K - 1...0$:
         A. if $\tau_a(k) = \tau_a^{\text{free}}$ then $k^{\text{e}} = k$
         B. $C_a(k) = \max\{0, (k^{\text{e}} - k) \cdot T - \tau_a^{\text{free}}\}$

$Q^{\text{in}}(t) \equiv \bar{q}t + n(0)$, where $n(0)$ is the occupancy of the link at time $t = 0$. A substitution of this in (3) yields $d\tau(t_1) \approx dn/\bar{q}$ and, when substituted in (4),

$$C(t_0) \approx dn/\bar{q} \cdot (Q^{\text{in}}(t^{\text{e}}(t_0) - \tau^{\text{free}}) - Q^{\text{in}}(t_0)). \tag{5}$$

This expression is straightforward to evaluated in a microsimulation context, where $dn = 1$ corresponds to the mass of a single agent and the difference in accumulated flows is easily evaluated by counting the agents leaving the considered link between t_0 and $t_{\text{e}}(t_0) - t_{\text{free}}$. A further simplification is obtained by replacing the accumulated flows in (5) by their linear approximations, which results for $dn = 1$ in

$$C(t_0) \approx t^{\text{e}}(t_0) - \tau^{\text{free}} - t_0. \tag{6}$$

An application of this result to a system optimal route assignment requires to calculate $C_a(t_0)$ for every link a and entry time t_0 in the network, and to add this term to the time-dependent link travel time that is evaluated in the route replanning of every agent. Alg. 2 outlines the arguably most straightforward implementation of this approach in a time-discrete multi-agent simulation.

4 Experimental results

This section presents the result of a simulation-based comparison of the two presented routing approaches. The simulation setup is based on a real-world evacuation scenario for the Indonesian city of Padang. Padang faces high risk of being inundated by a tsunami wave. The city has approximately 1,000,000 inhabitants, with more than 300,000 people living in the highly endangered area with an elevation of less then 10 m above see level. An overview map of the city is shown in fig. 1 (left). The area more than 10 m above sea level is assumed to be safe (in dark color). A detailed description of the evacuation scenario can be found in [12].

Two simulations are conducted: *Run 1* implements the Nash equilibrium approach described in sec. 3.1. *Run 2* implements the system optimal approach described in sec. 3.2. Both simulations run on a network with 6,289 nodes and 16,978 unidirectional links. The synthetic population consists of 321,281 agents.

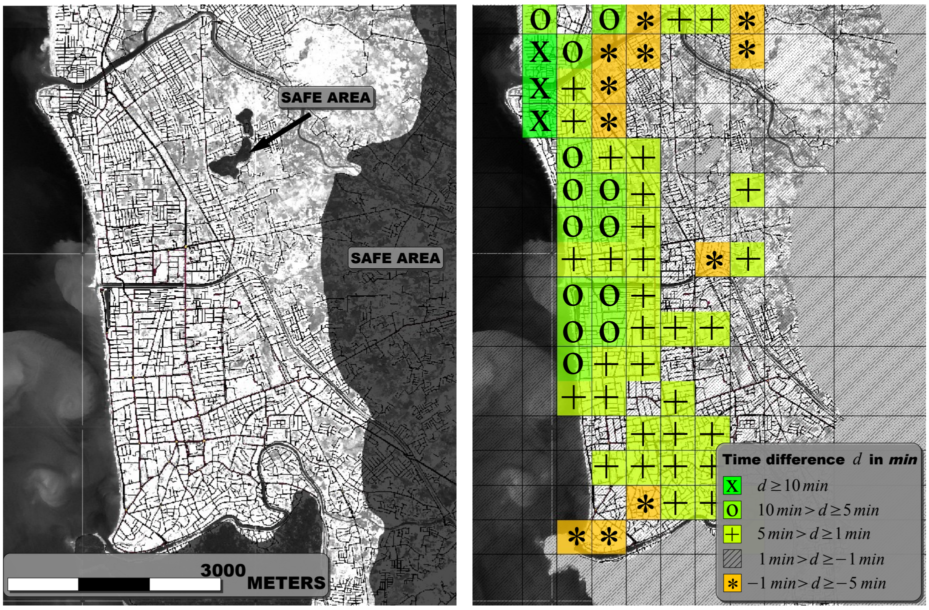

Fig. 1. Left: Overview map of downtown Padang. The safe area with an elevation of more than 10 m is in dark color, all other area is defined as unsafe. Right: Differences in evacuation time between Nash equilibrium approach and system optimal approach. In green parcels (X,O,+), the system optimal approach evacuates faster than the Nash approach, whereas red parcels (*) indicate the opposite.

This is the number of people living less than 10 m above see level. Both simulations are run for 200 iterations on a 3 GHz CPU running JAVA 1.5 on Linux. For *run 1* the overall runtime is 9:31 hours and for *run 2* it is 17:00 hours.

Fig. 2 (left) compares the learning progress of both approaches. In *run 1*, the average evacuation time per agent converges to 1718 seconds, and in *run 2* it converges to 1612 seconds[3]. This means that each agent gains on average 106 seconds in the system optimal approach. In both cases, the average evacuation time drops very fast in the first iterations, but from iteration 10 on it increases again. This effect is caused by the fact that in the first iterations not all agents manage to escape the tsunami, and agents that are caught in the flood wave are not considered in the calculation of the evacuation time. Since in the early iterations many agents starting in the coastal area with relatively long evacuation routes do not manage to escape, the average evacuation time is lower than during mid-iterations, where these agents have learned better evacuation routes.

Fig. 2 (right) compares the evacuation curves of *run 1* and *run 2* after 200 iterations of learning. The evacuation curve of *run 2* is steeper than the evacuation curve of *run 1*, which implies a higher outflow rate. The overall evacuation

[3] The smoothing of the learning curves after iteration 150 results from a deactivation of the router, such that in iterations 150–200 the agents only select from previously generated routes, which stabilizes the simulation.

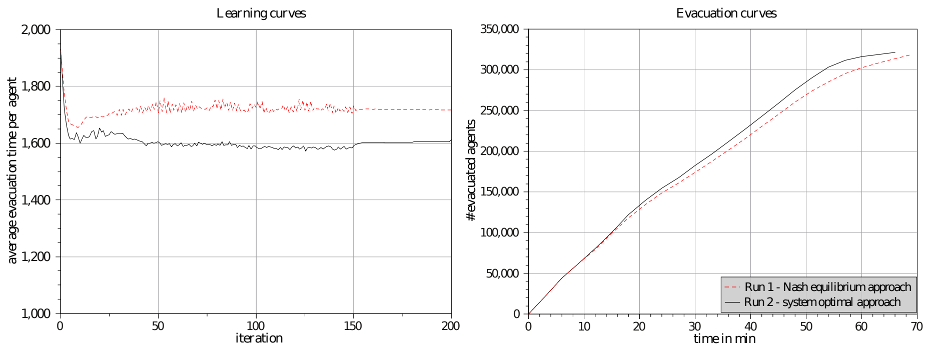

Fig. 2. Left: Average evacuation time per agent over the learning iteration number. Right: Comparison of the evacuation curves of *run 1* and *run 2*.

time of *run 2* is about 66 min, which is 3 min faster than in *run 1*. However, not all agents benefit from the system optimal approach. Fig. 1 (right) shows that mainly agents in the hinterland of Padang lose time in the system optimal approach, whereas many agents in the costal area of the city benefit by more than 10 min. This can be explained in the following way: The agents starting their trips in the hinterland are technically in front of the multi-link queues that spill back from the safe area to the coastal area. Consequently, they have the greatest effect on the total travel time. In system optimal conditions, the hinterland agents account for what they impose on the coastal agents by effectively giving way to them.

5 Conclusion and outlook

This article demonstrates that multi-agent simulations can be used to identify efficient evacuation strategies. Our results show that mathematically motivated cooperative routing solutions can be obtained with an acceptable computational overhead even in a purely simulation-based system. The presented cooperative routing approach, which approximates a system optimal solution, generates a substantially higher evacuation throughput than an alternative non-cooperative routing strategy. Even though the results from the system optimal approach do not reflect real human decision making, they serve as benchmark solutions and help to identify routing recommendations (e.g., placement of evacuation signs). The presented experiments with more than 300,000 evacuees show the feasibility of our approach even for large evacuation scenarios. Our ongoing research focuses on more precise system optimal routing strategies.

6 Acknowledgments

This project was funded in part by the German Ministry for Education and Research (BMBF), under grants numbers 03G0666E (“last mile”) and 03NAPI4 (“Advest”).

References

1. M. Ben-Akiva and S. R. Lerman. *Discrete choice analysis.* The MIT Press, Cambridge, MA, 1985.
2. M. Bierlaire, G. Antonini, and M. Weber. Behavioral dynamics for pedestrians. In K. Axhausen, editor, *Moving through nets: The physical and social dimensions of travel.* Elsevier, 2003.
3. X. Chen and F. Zhan. Agent-based modeling and simulation of urban evacuation: Relative effectiveness of simultaneous and staged evacuation strategies. Paper 04-0329, Transportation Research Board Annual Meeting, Washington, D.C., 2004.
4. J. Ferber. *Multi-agent systems. An Introduction to distributed artificial intelligence.* Addison-Wesley, 1999.
5. C. Gawron. An iterative algorithm to determine the dynamic user equilibrium in a traffic simulation model. *International Journal of Modern Physics C*, 9(3):393–407, 1998.
6. D. Helbing, I. Farkas, P. Molnar et al. Simulation of pedestrian crowds in normal and evacuation situations. In *Pedestrian and Evacation Dynamics*, Proceedings of the 1st international conference, Duisburg, 2001. Springer, 2002.
7. A. Hobeika and C. Kim. Comparison of traffic assignments in evacuation modeling. *IEEE Transactions on Engineering Management*, 45(2):192–198, 1998.
8. M. Jha, K. Moore, and B. Pashaie. Emergency evacuation planning with microscopic traffic simulation. Paper 04-2414, Transportation Research Board Annual Meeting, Washington, D.C., 2004.
9. H. Klüpfel, T. Meyer-König, A. Keßel et al.. Simulating evacuation processes and comparison to empirical results. In M. Fukui et al, editor, *Traffic and granular flow '01*, pages 449–454. Springer, Berlin Heidelberg New York, 2003.
10. E. Kwon and S. Pitt. Evaluation of emergency evacuation strategies for downtown event traffic using a dynamic network model. Paper 05-2164, Transportation Research Board Annual Meeting, Washington, D.C., 2005.
11. G. Lämmel, D. Grether, and K. Nagel. The representation and implementation of time-dependent inundation in large-scale microscopic evacuation simulations. *Transportation Research Part C: Emerging Technologies*, In Press, Corrected Proof:–, 2009.
12. G. Lämmel, M. Rieser, K. Nagel et al. Emergency preparedness in the case of a tsunami – evacuation analysis and traffic optimization for the Indonesian city of Padang. In *Pedestrian and Evacuation Dynamics*, Proceedings of the 4th international conference, Wuppertal, 2008. Springer, 2009.
13. Q. Lu, B. George, and S. Shekhar. Capacity constrained routing algorithms for evacuation planning: A summary of results. *LNCS*, 3633:291–307, 2005.
14. J. Nash. Non-cooperative games. *Annals of Mathematics*, 54(2):286–295, 1951.
15. S. Peeta and H. Mahmassani. System optimal and user equilibrium time-dependent traffic assignment in congested networks. *Annals of Operations Research*, 60:81–113, 1995.
16. S. Peeta and A. Ziliaskopoulos. Foundations of Dynamic Traffic Assignment: The Past, the Present and the Future. *Networks and Spatial Economics*, 1(3):233–265, 2001.
17. V. Schneider and R. Könnecke. Simulating evacuation processes with ASERI. In *Pedestrian and Evacation Dynamics*, Proceedings of the 1st international conference, Duisburg, 2001. Springer, 2002.
18. Y. Sheffi. *Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods.* Prentice-Hall, Englewood Cliffs, NJ, USA, 1985.