

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

Gregor Lämmel \* and Gunnar Flötteröd †

April 20, 2009

Report TRANSP-OR 090420  
Transport and Mobility Laboratory  
Ecole Polytechnique Fédérale de Lausanne  
`transp-or.epfl.ch`

---

\*Berlin Institute of Technology, Germany, [laemmel@vsp.tu-berlin.de](mailto:laemmel@vsp.tu-berlin.de)

†Transp-OR, Ecole Polytechnique Fédérale de Lausanne, Switzerland, [gunnar.floetteroed@epfl.ch](mailto:gunnar.floetteroed@epfl.ch)

## Abstract

Disaster and evacuation planning crucially depend 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.

This paper compares two routing strategies by simulating them in a real-world evacuation scenario (Birkmann et al., 2008; Lämmel et al., 2008b). Both strategies are generated by a learning-based multi-agent simulation: 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 unilateral deviation. 2. The system optimal approach, where the average travel time per agent is minimized. Here, learning agents are no longer optimizing their individual travel time only but in some way also take care about the others.

The presented "agent-based learning" approach to the realization of different routing strategies has its origin in traffic modeling, where it is known as the "dynamic traffic assignment (DTA) problem", routing strategy 1 is called "user optimal", and routing strategy 2 is denoted as "system optimal" (Peeta & Ziliaskopoulos, 2001). The added value of the agent-based approach is its natural representation of individual travelers by software agents, whereas classical DTA approaches are based on mathematical formulations that replace agents by continuous traffic streams. However, the improved modeling capabilities of multi-agent simulations come at the price of greater difficulties in their mathematical treatment. The agent-based routing approaches 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 the paper is organized as follows. Section 2 gives an overview of the dynamic simulation framework. In Section 3, the investigated routing strategies are described in detail. Section 4 presents simulation results based on which both strategies are evaluated. Finally, Section

5 concludes the article and provides an outlook on more advanced routing approaches that are currently under investigation.

## 2 Simulation framework

We implemented our experiments in the MATSim simulation framework (MATSIM [www page](#), accessed 2008). Since the details of this system are described elsewhere, e.g., (Lämmel et al., 2008b; Lämmel & Nagel, 2009)), only a brief description is given here.

MATSim always starts with a synthetic population, which is a randomly generated population of individuals that is based as much as possible on existing data such as census data. Every synthetic individual possesses one or several plans. These plans are possible realizations of the “intentions” of the synthetic individuals. 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 of every agent is selected for execution in a simulation of the physical world. The learning logic tests different plans, eventually discards poor plans, and sometimes generates new plans to be tested (Ferber, 1999).

The model of the real world is a pedestrian traffic flow simulation, where each street (link) is represented by a first-in/first-out queue with three parameters (Gawron, 1998): minimum link traversal time, maximum link outflow capacity (in evacuees per hour), and link space capacity (in evacuees). The link space capacity limits 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 (Lämmel et al., 2008a) describes in detail how this issue is resolved.

The route learning process relies heavily on a time-dependent Dijkstra algorithm that calculates best routes through a network where to every link a time-dependent cost function is attached.

In an evacuation setting, there typically exists no single destination node

for the routing because every node outside the evacuation area is a possible destination. To resolve this, the standard approach (e.g., (Lu et al., 2005)) is to extend the network in the following way: All links which lead out of the evacuation area are connected, using virtual links with infinite flow capacity and zero length, to a special “evacuation node”. Doing so, Dijkstra’s algorithm will always find the shortest route from any node inside the evacuation area to this evacuation node.

### 3 Routing solutions

Each agent iteratively optimizes its personal evacuation plan during a simulation run. After each iteration, every agent calculates the cost of the most recently executed plan. Based on this cost, the agents revise the most recently executed plans. Some agents generate new plans using the time-dependent router. The others select an existing plan they have previously used. This plan selection is realized as a Logit model that stabilizes the simulation dynamics by allowing slightly suboptimal plans to be (re)considered for execution as well.

In the following, we discuss two different cost functions that lead either to user optimal or the system optimal routing solutions. Note that we modify the agent’s routing behavior only by adjusting the costs based on which the routing is conducted, but we do not require to change the replanning logic itself.

#### 3.1 Nash equilibrium approach

In a Nash equilibrium, no agent can gain by unilateral deviation from its current evacuation plan (Nash, 1951). 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  $k = 0 \dots K - 1$ . The time-dependent link travel time when entering link  $a$  in time step  $k$  is denoted by  $\tau_a(k)$ . Alg. 2 drafts the Nash-equilibrium routing logic.

---

**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  $k$
- 

### 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 recalculate their routes needs to be replaced by the marginal travel time (Peeta & Mahmassani, 1995). The marginal travel time of a route is the amount by which the total system travel time changes if one additional agent chooses that route. It is the sum of the cost experienced by the added vehicle and the imposed on other vehicles. The latter is subsequently denoted as the "social cost". Since the marginal route travel time is link-additive, we derive an approximation only for a single link. The subsequent development is based on continuous quantities. 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  $\tau^{\text{free}}$  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  $t^e(t_0)$ . Now, consider another “affected” agent that enters the link at  $t_1 > t_0$ , and assume that this agent leaves the link before  $t^e(t_0)$ . 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:

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

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  $Q^{\text{in}}(t) \equiv Q^{\text{out}}(t) \equiv \bar{q}t$ . A substitution of this in (3) yields  $d\tau(t_1) \approx dn/\bar{q}$  and, when substituted in (4),

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

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  $t_e(t_0) - t_{\text{free}}$ . A further simplification is obtained by replacing the accumulated flows in (5) by their linear approximations, which results for  $dn = 1$  in

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

An application of this result to an approximately 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 route replanning of every agent. Algorithm 2 outlines the arguably most straightforward implementation of this approach in a multi-agent simulation.

---

**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.  $k^e = K$
      - ii. for  $k = K - 1 \dots 0$ :
        - A. if  $\tau_a(k) = \tau_a^{\text{free}}$  then  $k^e = k$
        - B.  $C_a(k) = \max\{0, (k^e - k) \cdot T - \tau_a^{\text{free}}\}$
- 

## 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 than 10 m above sea level. An overview map of the city is shown in fig. 1 (left). The area higher than 10 m above sea level is assumed to be safe and colored green in the map. A detailed description of the evacuation scenario can be found in (Lämmel et al., 2008b; Lämmel & Nagel, 2009).

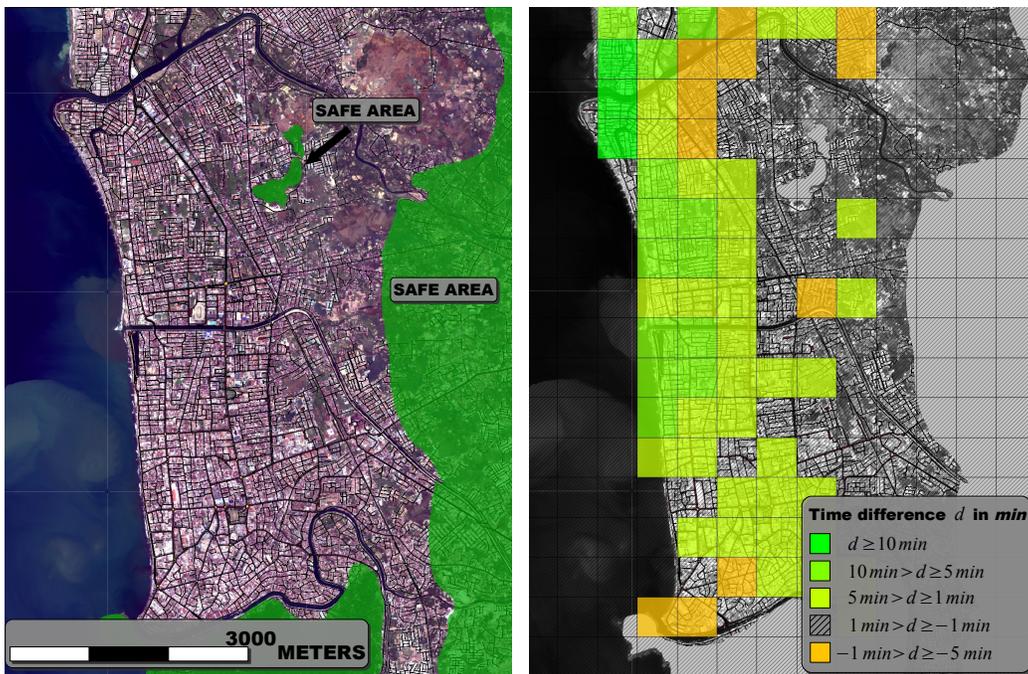


Figure 1: Left: Overview map of downtown Padang. The safe area with an elevation of more than 10 m is colored green, all other area is defined as unsafe. Right: Differences in evacuation time between Nash equilibrium approach and system optimal approach. In green parcels, the system optimal approach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

Two different runs are conducted: *Run 1* implements the Nash equilibrium

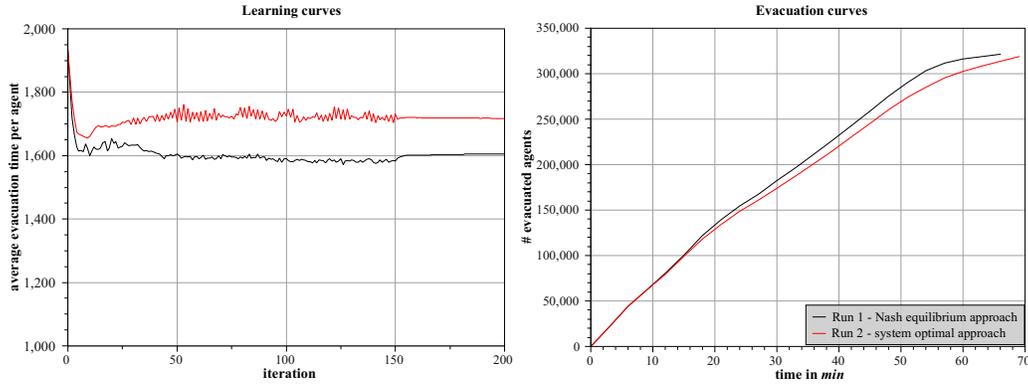


Figure 2: Left: Average evacuation time per agent over the learning iteration number. Upon convergence, the system optimal approach performs 106 seconds better than the Nash equilibrium approach. Right: Comparison of the evacuation curves of *run 1* and *run 2*. *Run 2* generates a steeper gradient (higher outflow rate) than *run 1* and also an about 3 min shorter overall evacuation time.

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. This is the number of people living less than 10 m above sea 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 was 9:31 hours and for *run 2* 17:00 hours. The system optimal routing is computed roughly half as fast a Nash solution because of the more complex calculations that have to be performed to calculate the social costs. However, the current implementation of the system optimal approach is not optimized for runtime performance and is likely to be improved in future versions.

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. This means that in the system optimal approach, each agent gains on average 106 seconds. 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, agents that caught in the floodwave are not considered for the evacuation time calculation. Since in the early iterations many agents starting in the coastal area (longer evacuation routes) do not manage to escape, the average evacuation time is lower than during mid-iterations, where these “costal 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 time of *run 2* is about 66 min, which is 3 min faster than in *run 1*. Not all agents gain by applying the marginal cost approach. For example, some agents may make detours in order clear some roads for other agents. 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.

## 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 is an approximation of the "system optimal routing" well known in the field of dynamic traffic assignment, generates a substantially higher evacuation throughput than an alternative non-cooperative routing strategy. The presented experiments with more than 300,000 evacuees show the feasibility of our approach even for large evacuation scenarios.

Our current work focuses on a less approximate computation logic for the social costs required in the system optimal approach. This approach is likely to have some similarities with Reinforcement Learning (Sutton & Barto, 1998).

## 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

- J. Birkmann, et al. (2008). ‘Numerical Last-Mile Tsunami Early Warning and Evacuation Information System (“Last-Mile – Evacuation ”)’. In *GEOTECHNOLOGIEN Science Report No. 10: “Early Warning Systems in Earth Management”*, Osnabrück.
- J. Ferber (1999). *Multi-agent systems. An Introduction to distributed artificial intelligence*. Addison-Wesley.
- C. Gawron (1998). ‘An Iterative Algorithm to Determine the Dynamic User Equilibrium in a Traffic Simulation Model’. *International Journal of Modern Physics C* 9(3):393–407.
- G. Lämmel, et al. (2008a). ‘The representation and implementation of time-dependent inundation in large-scale microscopic evacuation simulations’. *VSP Working Paper* (08-20).
- G. Lämmel & K. Nagel (2009). ‘Multi agent based large-scale evacuation simulation’. Paper 09-2135, Transportation Research Board Annual Meeting, Washington, D.C.
- G. Lämmel, et al. (2008b). ‘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, Berlin. Springer.
- Q. Lu, et al. (2005). ‘Capacity Constrained Routing Algorithms for Evacuation Planning: A Summary of Results’. *LNCS* 3633:291–307.

- MATSIM www page (accessed 2008). 'MultiAgent Transport SIMulation'.  
<http://matsim.org/>.
- J. Nash (1951). 'Non-Cooperative Games'. *Annals of Mathematics* 54(2):286–295.
- S. Peeta & H. Mahmassani (1995). 'System optimal and user equilibrium time-dependent traffic assignment in congested networks'. *Annals of Operations Research* 60:81–113.
- S. Peeta & A. Ziliaskopoulos (2001). 'Foundations of Dynamic Traffic Assignment: The Past, the Present and the Future'. *Networks and Spatial Economics* 1(3):233–265.
- R. Sutton & A. Barto (1998). *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA.