Bottlenecks and Congestion in Evacuation Scenarios: A Microscopic Evacuation Simulation for Large-Scale Disasters

Gregor Lämmel
Transport Systems Planning and Transport Telematics (VSP)
TU Berlin, Salzufer 17-19, Sekr. SG 12, 10587 Berlin, Germany
laemmel@vsp.tu-berlin.de

Marcel Rieser
Transport Systems Planning and Transport Telematics (VSP)
TU Berlin, Salzufer 17-19, Sekr. SG 12, 10587 Berlin, Germany
and
Institute for Transport Planning and Systems (IVT)
ETH Zurich, 8093 Zurich, Switzerland
rieser@vsp.tu-berlin.de

Kai Nagel
Transport Systems Planning and Transport Telematics (VSP)
TU Berlin, Salzufer 17-19, Sekr. SG 12, 10587 Berlin, Germany
nagel@vsp.tu-berlin.de

ABSTRACT

Multi Agent Simulation has increasingly been used for transportation simulation in recent years. With current techniques, it is possible to simulate systems consisting of several million agents. Such Multi Agent Simulations have been applied to transportation simulation for whole cities and even large regions. In this paper we demonstrate how to adapt an existing multi agent transportation simulation framework to large-scale pedestrian evacuation simulation. The underlying flow model simulates the traffic based on a simple queue model where only free speed and bottleneck capacities are taken into account. The queue simulation, albeit simple, captures the most important aspects of evacuations such as the congestion effects of bottlenecks and the time needed to evacuate the endangered area.

During the simulation, each evacuee optimizes his/her personal evacuation route to find the fastest escape route. At this point two different routing solutions are considered: (1) An "empty network" routing solution, where every evacuee follows the path that would be fastest in an empty network. (2) A "Nash equilibrium" approach, where, via iterations every evacuating person attempts to find a route that is optimal for him/herself under the given circumstances. Both approaches can be considered as benchmarks: the first as one where congestion effects are not taken into account in the path choice; the second one as one which might be achieved by appropriate training or guidance while maintaining acceptability in the sense that no person could gain by deviating from this solution. The results from the simulation give an estimate of the time it could take to evacuate the endangered area. We applied the system to a hypothetical scenario, namely a dam-break of the Sihlsee dam near Zurich, which would lead to an inundation of large parts of the city of Zurich within two hours. We show how well both approaches perform with respect to evacuation time and the outflow rate of evacuees.

Cite as: Title, Author(s), *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, Padgham, Parkes, Müller and Parsons (eds.), May, 12-16., 2008, Estoril, Portugal, pp. XXX-XXX.

Keywords

multi agent simulation, large-scale evacuation simulation

1. INTRODUCTION

The evacuation of whole cities or even regions is an important problem, as demonstrated by recent events such as the evacuation of Houston in the case of Hurricane Rita or the evacuation of coastal cities in the case of Tsunamis. A robust and flexible simulation framework for such large-scale disasters helps to predict the evacuation process. Furthermore, it is possible to recognize bottlenecks in advance, so that an elimination of those bottlenecks is possible. This should lead to a better preparedness for an event of evacuation for cities or regions that face a high risk of natural disasters.

2. RELATED WORK

Disaster and evacuation planning has become an important topic in science and politics. In principle there are two different situations: evacuation of buildings, ships and airplanes or the like on the one hand, or evacuation of whole cities or even regions on the other hand. The former involves normally the evacuation of pedestrians, where the latter is rather associated with the evacuation by car.

In the area of pedestrian evacuation simulation, there has been done considerable research in the last 20 years. A good overview about models and software for pedestrian evacuation simulation can be found in the proceedings of the conference "Pedestrian and Evacuation Dynamics" [35, 9, 10]. Corresponding to the two different types of problems, there are two different basic approaches for simulating the traffic flow:

(1) Methods of dynamic traffic assignment (DTA) have been applied to evacuation simulation on the city or regional scale. Some examples are: MITSIM [19], DYNASMART [22] or VISSIM [14]. The DTA approach is based on the analogy between traffic and hydrodynamic characteristics of fluids. That means DTA is a macroscopic approach and reduces the problem of evacuation dynamics to a well known physical problem. On state of the art hardware it is possible to handle even large-scale scenarios with this approach. – However, in DTA it is not straightforward to deal with the inhomogeneity of a population. For this, a microscopic simulation is needed, where all people are simulated as individuals.

(2) Microscopic simulations are often based on Cellular Automata (CA) [28, 29, 16]. In CA models each evacuee is designed as an individual; therefore it is possible to simulate also population structures where people have different speeds or ranges, or more complex behavior. The modeling of complex behavior in evacuation simulation has become important in recent years. People could for example ignore warnings or might not choose the nearest emergency exit, furthermore people tend to follow others (herd behavior) [15, 23]. Agent oriented research groups have modeled such behavior [27, 30]. In general it is expected that complex behavior leads to longer evacuation times, consequently a simulation that ignores such behavior patterns is probably optimistic.

The aim of this approach is to develop a simulation framework for large-scale scenarios, e.g. for large cities with a population of hundreds of thousands. A standard CA-based approach is not applicable here, because the area of those cities could be several hundred square kilometers. In this case a CA-model would consist of more than 10^9 cells, leading to rather long computing times.

In contrast, a DTA approach, as pointed out earlier, is not able to handle complex individual behavior. One possible approach to deal with such large-scale scenarios but to retain persons as individual agents is based up on a modified queuing model [11, 36]. The queuing model simplifies streets to edges and crossings to nodes; the difference to standard queuing theory is that agents (particles) are not dropped but spill back, causing congestion. This graph-oriented model is defined by lengths/widths, free speed and flow capacity of the edges. This simplification leads to a major speedup of the simulation while keeping results realistic. For example, the simulation of the whole (motor) traffic of Switzerland (approx. 5 million trips) takes less then 5 minutes for 24h real time [32]. In this work the adaptation of the existing multi agent transportation simulation framework to large-scale pedestrian evacuation simulation is described.

3. MULTI AGENT SIMULATION

Our simulation is constructed around the notion of agents that make independent decisions about their actions. In this case study, each evacuee is modeled as an individual agent in our simulation. In the simulation the agents try to find the best (in terms of time) escape route, whereby the real world is modeled as a network constructed of nodes (intersections) and links (roadway between intersections). The overall approach consists of three important pieces:

- Each agent independently generates a so-called *plan* which encodes its intended escape route.
- All agents' plans are simultaneously executed in the simulation of the physical system. This is also called the *traffic flow simulation* or *mobility simulation*.
- There is a mechanism that allows agents to *learn*. In our implementation, the system iterates between plans generation and traffic flow simulation (i.e. systematic relaxation [20, 4]). The system remembers several plans per agent, and scores the performance of each plan. Agents normally chose the plan with the highest score, sometimes re-evaluate plans with bad scores, and sometimes obtain new plans. Further details will be given below.

The simulation approach is the same as in many of our previous papers (e.g. [33, 3]) on the same subject. The results of this paper are based on a re-implementation of the MATSim framework in Java [25]. Since not all elements of MATSim are important for an evacuation simulation, the following exposition is a shortened and simplified description of key elements.

A **plan** contains the itinerary of activities the agent wants to perform during the day, plus the intervening trips the agent must take to travel between activities. An agent's plan details the order, type, location, duration and other time constraints of each activity, and the mode, route and expected departure and travel times of each trip. This paper concentrates on "home" and "evacuated" as the only activities, and "walk" as the only mode.

A plan can be modified by the **router module**: The router is implemented as a time-dependent Dijkstra algorithm. It calculates link travel times from the output of the traffic flow simulation. The link travel times are encoded in variable-sized time bins, so they can be used as the time-dependent weights of the links in the network graph.

The **traffic flow simulation** executes all agents' plans simultaneously on the network. In the work presented here, the plans contain the departure time and the exact routes, and agents just follow these prescriptions; learning is implemented via iterations (see below). The traffic flow simulation is implemented as a queue simulation, where each street (link) is represented as a FIFO (first-in first-out) queue with three restrictions [11, 5]. First, each agent has to remain for a certain time on the link, corresponding to the free speed travel time. Second, a link flow capacity is defined which limits the outflow from the link. If, in any given time step, that capacity is used up, no more agents can leave the link. Finally, a link storage capacity is defined which limits the number of agents on the link. If it is filled up, no more agents can enter this link. The traffic flow simulation provides output describing what happened to each individual agent during the execution of its plan.

The outcome of the traffic flow simulation (e.g. congestion) depends on the planning decisions made by the decision-making modules (in this case, the router). However, those modules can base their decisions on the output of the traffic flow simulation (e.g. knowledge of congestion) using **feedback** from the multi-agent simulation structure [20, 4]. This sets up an iteration cycle which runs the traffic flow simulation with specific plans for the agents, then uses the planning modules to update the plans, these changed plans are again fed into the traffic flow simulation, etc., until consistency between modules is reached.

The feedback cycle is controlled by the **agent database**,

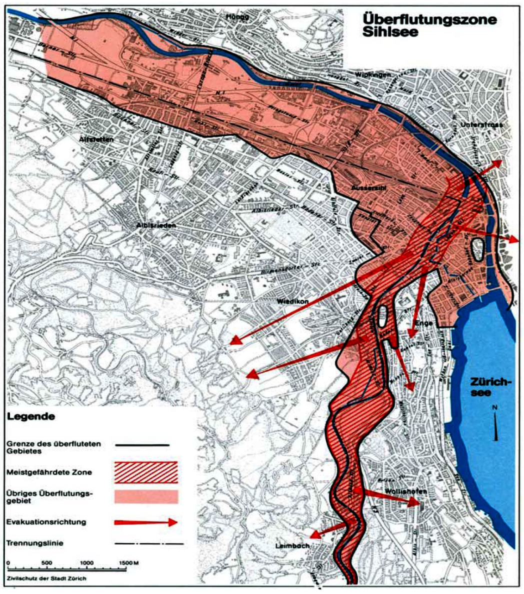

Figure 1: Inundation map provided by the Zurich civil defense office

Figure 2: Empty evacuation network

which also keeps track of multiple plans generated by each agent, allowing agents to reuse those plans at will. The repetition of the iteration cycle coupled with the agent database enables the agents to learn how to improve their plans over many iterations. This circle continues until the system has reached a relaxed state. At this point, there is no quantitative measure of when the system is "relaxed"; we just allow the cycle to continue until the outcome seems stable. The actual number of iterations that are needed depends on the scenario. Normally 100 to 200 iterations are sufficient to reach this "relaxed" state (also see below).

In order to compare plans, it is necessary to assign a quantitative **score** to the performance of each plan. In principle, arbitrary scoring schemes can be used (e.g. prospect theory [2]). In this work it is assumed that the agents are only interested in minimizing their individual evacuation time. For that reason, the utility of a plan is just the negative of the time needed to reach the safe area.

4. SCENARIO

A hypothetical event of a dam-break of the Sihlsee dam was chosen. This would lead to an inundation of parts of the city of Zurich. According to the civil defense office there will be an advance warning time of about 110 minutes until the inundation will reach the city center. The civil defense office also provides an instruction sheet [1] with an inundation map of the area at risk (shown in figure 1).

4.1 Data Basis

There are two main inputs that have to be provided to the simulation framework. At first the simulation needs a network. We extracted the evacuation network by projecting the inundation map from the civil defense office onto the network of Switzerland provided by NAVTEQ[1]. This extraction has been done semi-automatically. First, all boundary nodes were selected manually, and after this all links and nodes inside the so selected area were selected automatically. So the overall effort of pre-processing was manageable. The original NAVTEQ network of Switzerland consists of about 400k nodes and 880k links. After cropping, the resulting network consist of 3037 nodes and 6120 links. It is shown in figure 2.

The other important input to the simulation framework is a so-called "plans file", containing information about people and their plans, including home and work locations. A synthetic population for the area of Zurich was generated by [26] and provided to us. The population was generated using data from the Swiss census for the year 2000 [12] and information about facilities in the city center. Every person in this synthetic population obtains one complete day plan, describing all activities the person performs during a day. The first work location appearing in a plan of each agent was extracted to build the agents' initial locations for the evacuation. In the end, there were 165571 agents with a work activity within the endangered area. This set of agents and locations builds our start setup for the evacuation; this means we will simulate a break of the Sihlsee dam during regular working hours.

4.2 Calibration of the Queuing Model

Since the underlying simulation framework is mainly designed for the simulation of motorized transportation, several adaptations are necessary. At first it is obvious that the evacuees do not care about the traffic direction. So we allowed all links in the street network to be used in both directions. Given the link length, the queuing model is described by three parameters for each link. The parameters are: **flow capacity**, **storage capacity** and **free flow speed**. These parameters had to be calibrated to achieve an appropriate flow dynamic for pedestrians. In literature the flow dynamic of pedestrians is often described by fundamental diagrams [37, 31]. These diagrams show the velocity as a function of

[1]NAVTEQ is a provider of digital maps for in-vehicle navigation systems (see also http://www.navteq.com/)

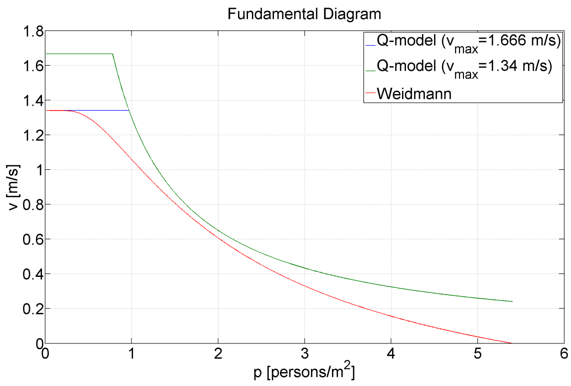

Figure 3: Weidmann's fundamental diagram compared to queuing model

the density of pedestrians. Weidmann pointed out that the relation between density and velocity is adequately captured by the so-called Kladek-formula [37][2]:

$$v_{F,hi}(D) = v_{F,hf} \times [1 - e^{-\gamma \times (\frac{1}{D} - \frac{1}{D_{max}})}]$$

With:

- $v_{F,hi}$ the velocity at a particular density $[m/s]$,
- $v_{F,hf}$ the velocity at free flow $[m/s]$,
- γ a free parameter $[persons/m^2]$,
- D the actual density $[persons/m^2]$ and
- D_{max} the density at which no flow occurs $[persons/m^2]$.

Empirical studies showed the best results with $\gamma = 1.913$ $persons/m^2$, $v_{F,hf} = 1.34$ m/s and $D_{max} = 5.4$ $person/m^2$.

Our queuing model, however, generates a speed-density relationship of the form $v = \min[v_{max}, 1/D]$ [36]. Therefore a complete agreement is not possible. However, as shown in figure 3, the flow dynamic produced by our queue model is not too far away from Weidmann's fundamental diagram. The details of the calibration are explained in the following paragraphs.

As the above mentioned NAVTEQ network is designed for transport simulation, we had to adjust the networks parameters accordingly. In the original network file, there is only information about number of lanes but not the width of the street, so we had to estimate it. According to the handbook Strassenprojektierung [6] the lane width on streets in Switzerland has to be 2.20-3.00 m for automobiles and 3.10-3.90 m for trucks. Taking this information, we set the width of all lanes in the network to 3.50 m. For pedestrian evacuation the flow capacity is assigned in persons per meter per second, but it depends on the actual density of persons. According to Weidmann [37] the maximum flow is about 1.3 $persons/(m \cdot s)$ at a density of 2 $persons/m^2$. The SFPE Handbook of Fire Protection Engineering [8] supports these values. Together with the lane width we got the flow capacity of 4.55 $persons/(lane \cdot s)$.

[2]Newer studies [34] imply other fundamental diagrams then those from Weidmann or Predtetschenski and Milinski. An adaptation of these values could, in consequence, become necessary in future.

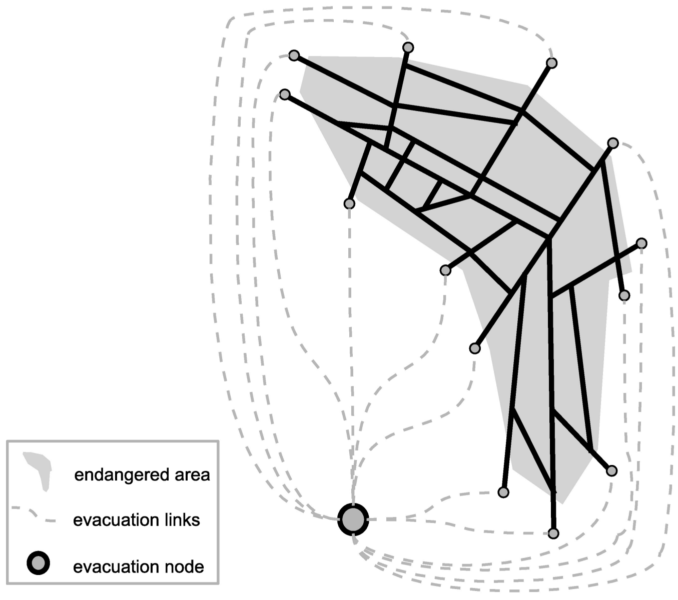

Figure 4: Sketch of the modified evacuation network

Another parameter for the queue simulation is the storage capacity of the links. Not to contradict the flow rate in Weidmann's fundamental diagram we set the storage capacity to 2 $persons/m^2$.

The free flow speed was set to $1.666 m/s$. This value is slightly higher then the $1.34 m/s$ recommended in literature, but the values presented by Weidmann reflect the pedestrian flow under normal conditions and not in a case of emergency. Before we can apply this approach to "real world" scenarios we have to verify all parameters and check if they are realistic.

Overall, there are 101 links that lead out of the evacuation area. Most of them have one or two lanes. The aggregated capacity of all these "escape links" is 787.15 $persons/s$. However, this capacity is a theoretical value since it is unlikely that the evacuees will find a way to distribute themselves in such a smooth way over the network. Rather it is expected that this outflow rate has a much lower value at the initial iteration, where all evacuees proceed on the assumption that the network is empty and there is free speed on all links. With the optimization of the evacuation procedure the outflow rate is expected to increase as the evacuees will make better use of all roads.

4.3 Initial Routing

Initial plans use the shortest path (according to free speed travel time) out of the evacuation area for all agents. Within the MATSim framework a shortest path router based on Dijkstra's shortest path algorithm [7] has been implemented. This router finds the shortest path in a weighted graph from one node to any other, whereby the actual weights for a link are defined by a time-dependent cost function. Since we want to evacuate the city as fast as possible, the weights represents the (expected) travel time[3].

There is, however, no particular node as the target of the

[3]For the initial evacuation plans the expected travel time is determined by free travel speed.

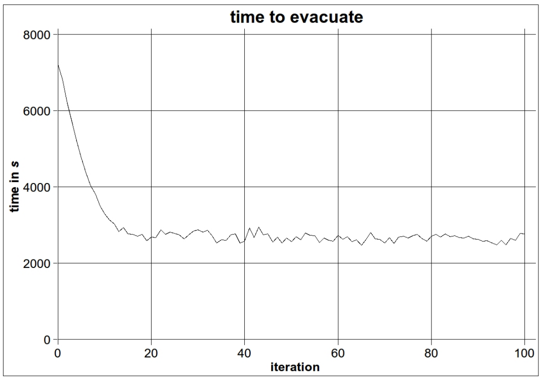

Figure 5: Evacuation time vs. iteration number

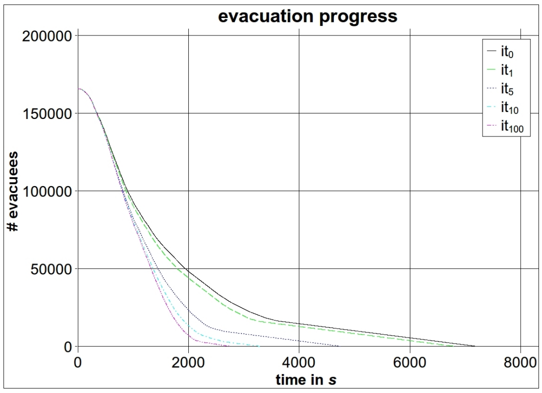

Figure 6: Evacuation progress

shortest path calculation, as the evacuees have more than one safe place to run to. Instead, in the underlying domain every node outside the evacuation area is a possible destination for an agent that is looking for an escape route. To resolve this, the standard approach (e.g. [24]) 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" (see figure 4). Doing so, Dijkstra's algorithm will always find the shortest route from any node inside the evacuation area to this evacuation node.

4.4 Re-Planning and Learning

At the end of each iteration, every agent scores the performed plan. In this study the scoring function is simply the negative of the travel time. This score is then memorized for the plan. After an agent has updated the score of its actual plan, it will be selected with a probability of 10% for re-routing. This replanning probability is a configurable parameter; 10% is a good compromise between slow convergence on the one hand, and over-reaction of the system on the other hand. In the re-routing procedure, the Dijkstra router is again applied to find the fastest escape route for the particular agent. The difference to the initial routing is that the weights for the links are no longer based on free speed travel times but on the experienced travel times from the last iteration. The travel times of all links are recorded and averaged into time bins. More precisely, the link traversal times of all pedestrians *entering* a link during a specific time bin are averaged. Those link travel times are then used when, during the Dijkstra computation, a specific link is entered by the algorithm. More details can be found in [18].

The size of these bins is configurable; for the present study, a size of 15 mins was used. If no traffic for a particular time bin and link occurs, free speed travel time is assumed for this time bin and link.

For agents that have not been chosen for re-planning, the plans with the highest scores (i.e. the plan with the fastest escape route) are selected for the next iteration. Repeating this iteration cycle, the agent behavior will move towards a Nash equilibrium. If the system were deterministic, then a state where every agent uses a plan that is a best response to the last iteration would be a fixed point of the iterative dynamics, and at the same time a Nash Equilibrium since no agent would have an incentive to unilaterally deviate. Since, however, the system is stochastic, this statement does not hold, and instead we look heuristically at projections of the system such as in Fig.5. In all such plots, 100 iterations is more than enough to arrive at a horizontal line, indicating that the iterative dynamics has reached a steady state.

In most (but not all) evacuation situations, the Nash equilibrium leads to a shorter overall evacuation time than when everybody moves to the geographically nearest evacuation point. On the other hand, a Nash equilibrium means that nobody has an incentive to deviate. The Nash equilibrium in an evacuation situation can therefore be considered as a solution that could be reached by appropriate training.

5. RESULTS

The simulation run was performed on a dual core CPU at 2.33 GHz with 2 GB of RAM. The computer runs JAVA jdk1.5_012 on Linux. The evacuation simulation was stopped after 100 re-planning cycles. The average runtime for an iteration was 123 seconds and the overall runtime was 3 hours and 24 minutes. The simulation consumed up to 1393MB of RAM. Besides the evacuation time, the outflow rate of the evacuation area has been recorded, too.

As expected, the evacuation time decreases significantly with the iterations. Especially within the early iterations, it drops very fast. A diagram that represents this process is shown in figure 5. The evacuation takes 7205 seconds at the initial iteration. Beginning with iteration 15 there are only small changes and it fluctuates randomly around 2676 seconds.

These values show only how long the overall evacuation takes but it tells nothing about the evacuation process itself. Therefore we evaluated the evacuation process for iteration 0, 1, 5, 10 and 100 in detail. Figure 6 shows the results.

The initial iteration results in a steep gradient (high outflow) at the beginning but it flattens very fast. As the iterations progress the initial gradient gets even steeper and becomes more linear.

Some statistics of the outflow of evacuees for the discussed iterations are given in table 1. Overall the results are as expected: both the maximum flow and the median flow are increasing with the iterations. Nevertheless, there are some interesting details. One interesting aspect is the low value for the median of the initial or 5th iteration. A possible reason for this phenomenon is that many agents try to perform

Iteration	max	mean	median
0	127	22.98	5
1	129	24.32	5
5	139	34.78	10
10	139	50.26	46
100	148	59.94	68

Table 1: Statistics of the outflow rate ($persons/s$)

the same escape route. If this happens they will line up in a few long queues, which will result in a low constant outflow rate.

The comparison of the snapshots for iteration 0 and 100 in figure 7 supports this hypothesis. Both snapshots were taken after 30 minutes of evacuation. The escape directions are indicated by black arrows. In iteration 0 there are considerably more evacuees at this point then in iteration 100. In the latter, the agents take advantage of six evacuation points (indicated by the red circles). This is much more effective then the behavior in the initial iteration, where only four evacuation points are used. Bottlenecks can also be detected. Figure 8 depicts this issue. In this figure the links are colored dependent on congestion. A green color indicates that the agents travel with free flow speed and as the color moves to red the flow speed decreases. It is not surprising that these congestion instances emerge at bridges, but the snapshot is taken after 100 iterations of learning and that means: there seems to be no better solution for the individual agent then to queue up on these bridges.[4]

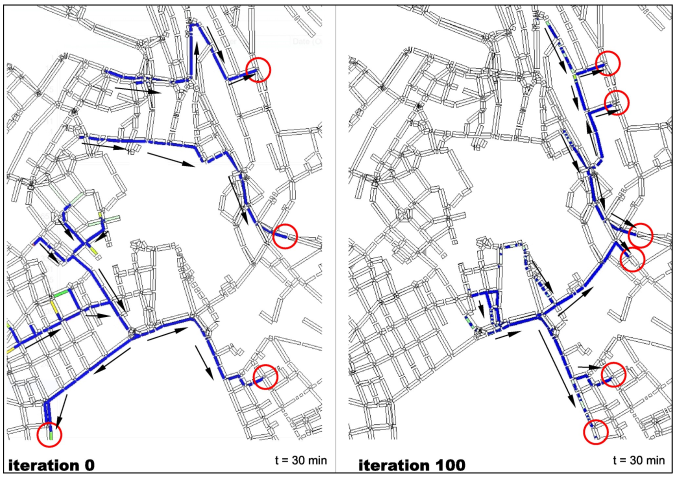

Figure 7: Comparison of two snapshots

6. DISCUSSION

The simulations concentrate on two types of agent behaviors: One where every agent follows the shortest path to the safe area; one where a Nash equilibrium is reached. Both can be considered as benchmarks:

- The first as one where agents are rational about their path choice, but unaware of congestion effects.
- The second as a solution that could be reached by training, assuming that agents follow the training solution also in the real situation.

Clearly, both can only be considered as benchmark solutions. In panic situations, people tend to be irrational and to display herd behavior [15]. Still, if even the Nash equilibrium solution does not leave enough time, then this would be a strong indicator that major measures would need to be taken to rectify the situation.

It should also be stated that Nash equilibrium and system optimum do not need to coincide – i.e. that solutions even better than the Nash equilibrium might be possible. Such solutions would, however, be unstable in the sense that people would have an incentive to deviate. Such solutions seem even more improbable than Nash equilibrium solutions.

Finally, one should mention that MATSim already contains the first hooks towards en-route replanning [17]. This would allow to add situation-based behavior into the simulation.

[4]For those who know the area: Since this preliminary study is based on a vehicular traffic network, it ignores links which can be used by pedestrians only. This could be corrected by using different network data.

Another issue concerns the mode choice: The investigation assumes that all evacuation is done by foot while it might be reasonable to assume that some people use cars or cycles, and they might even leave vehicles in the street to continue on foot if progress by vehicle becomes too slow. For the time being, such issues are not considered. The queue model could, to a certain extent, be parameterized to deal with mixed traffic, as long as all modes move with the same speed. Beyond that, one would arguably need to switch to a true two-dimensional model such as [15] or [21]. Such models could still operate on networks [13].

7. CONCLUSIONS

We introduced a microscopic pedestrian simulation framework for large-scale evacuations. It is implemented as a Multi Agent Simulation, where every agent tries to optimize its individual evacuation plan in an iterative way. The simulation framework is demonstrated through a case study based on a hypothetical dam-break of the Sihlsee dam near Zurich. Despite the underlying behavioral model being quite simple, the simulation gives plausible results regarding the predicted evacuation time and bottlenecks. The runtime performance shows that this approach is well suited for large scale scenarios. With state of the art hardware it is no problem to simulate much larger scenarios with over one million agents. In future work it is planed to apply this framework to an evacuation simulation in the case of a Tsunami warning for the Indonesian city of Padang. The improvement of the behavioral model (e.g. herd behavior [15] modified for large-scale scenarios [13]) could also be a topic of future work.

8. ACKNOWLEDGMENTS

This project was funded in part by the German Ministry for Education and Research (BMBF), under grant number 03G0666E ("last mile").

9. REFERENCES

[1] *Wasseralarm Sihlsee*. Zivilschutz Stadt Zürich (www.stadt-zuerich.ch). handout in German.

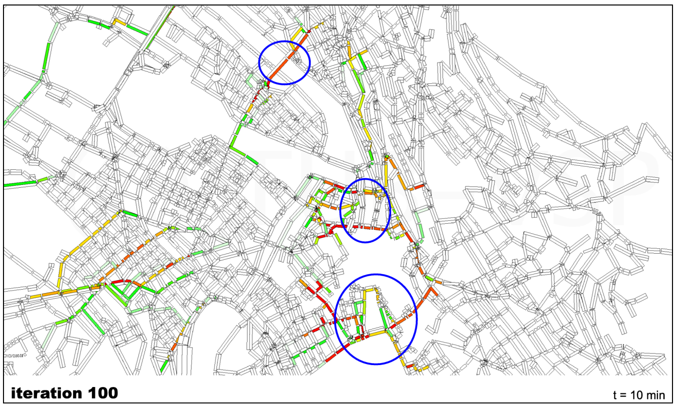

Figure 8: Bottlenecks after 100 iterations

[2] E. Avineri and J. Prashker. Sensitivity to uncertainty: Need for paradigm shift. Paper 03-3744, Transportation Research Board Annual Meeting, Washington, D.C., 2003.

[3] M. Balmer, B. Raney, and K. Nagel. Adjustment of activity timing and duration in an agent-based traffic flow simulation. In H. Timmermans, editor, *Progress in activity-based analysis*, pages 91–114. Elsevier, Oxford, UK, 2005.

[4] J. Bottom. *Consistent anticipatory route guidance*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 2000.

[5] N. Cetin, A. Burri, and K. Nagel. A large-scale agent-based traffic microsimulation based on queue model. In *Proceedings of Swiss Transport Research Conference (STRC)*, Monte Verita, CH, 2003. Earlier version, with inferior performance values: Transportation Research Board Annual Meeting 2003 paper number 03-4272.

[6] K. Dietrich. *Strassenprojektierung*. Schriftenreihe des IVT. Institute for Transport Planning and Systems ETH Zürich, 9 edition, 1998. in German.

[7] E. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269 – 271, 1959.

[8] P. DiNenno, editor. *SFPE Handbook of Fire Protection Engineering*. National Fire Protection Association, Boston, 2nd edition, 1995.

[9] E. R. Galea, editor. *Pedestrian and Evacuation Dynamics*. Proceedings of the 2nd international conference, London, 2003. CMS Press, University of Greenwich, UK, 2003.

[10] P. Gattermann, N. Waldau, and M. Schreckenberg, editors. *Pedestrian and Evacuation Dynamics*. Proceedings of the 3rd international conference, Vienna. Springer, Berlin, 2006.

[11] C. Gawron. An iterative algorithm to determine the dynamic user equilibrium in a traffic simulation model. In D. Wolf and M. Schreckenberg, editors, *Traffic and granular flow'97*, pages 469–474. Springer, Berlin, 1998.

[12] U. Germann. *Abschlussbericht zur Volkszählung 2000*, volume 1 of *Statistik der Schweiz*. Bundesamt für Statistik, Neuchâtel, 2005. in German.

[13] C. Gloor, P. Stucki, and K. Nagel. Hybrid techniques for pedestrian simulations. In *Cellular automata, Proceedings*, number 3305 in Lecture Notes in Computer Science, pages 581–590. Springer, 2004.

[14] L. Han and F. Yuan. Evacuation modeling and operations using dynamic traffic assignment and most desirable destination approaches. Paper 05-2401, Transportation Research Board Annual Meeting, Washington, D.C., 2005.

[15] D. Helbing, I. Farkas, and T. Vicsek. Simulating dynamical features of escape panic. *Nature*, 407:487–490, 2000.

[16] S. Hoogendoorn, P. Bovy, and W. Daamen. Microscopic pedestrian wayfinding and dynamic modelling. In Schreckenberg and Sharma [35], pages 123–154.

[17] J. Illenberger, G. Flötteröd, and K. Nagel. Enhancing matsim with capabilities of within-day re-planning. In *IEEE Intelligent Transportation Systems Conference*, Seattle, WA, 2007.

[18] R. R. Jacob, M. V. Marathe, and K. Nagel. A computational study of routing algorithms for realistic transportation networks. *ACM Journal of Experimental Algorithms*, 4(1999es, Article No. 6), 1999.

[19] 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.

[20] D. E. Kaufman, K. E. Wunderlich, and R. L. Smith. An iterative routing/assignment method for anticipatory real-time route guidance. Technical Report IVHS Technical Report 91-02, University of Michigan Department of Industrial and Operations Engineering, Ann Arbor MI 48109, May 1991.

[21] H. Klüpfel. The simulation of crowd dynamics at very large events – Calibration, empirical data, and validation. In Gattermann et al. [10].

[22] 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.

[23] C. Lieberman, D. Kurowski, M. Avila, L. Ricci, N. Thomas, C. Collyer, and B. Aguirre. Conceptual framework for simulating the pedestrian evacuation behavior from buildings. Paper 05-2297, Transportation Research Board Annual Meeting, Washington, D.C., 2005.

[24] Q. Lu, B. George, and S. Shekhar. *Capacity Constrained Routing Algorithms for Evacuation Planning: A Summary of Results*, volume 3633 of *Lecture Notes in Computer Science*, pages 291–307. Springer Berlin / Heidelberg, 2005.

[25] MATSIM www page. MultiAgent Transport SIMulation, accessed 2008.

[26] K. Meister, M. Rieser, F. Ciari, A. Horni, M. Balmer, and K. Axhausen. Anwendung eines agentenbasierten Modells der Verkehrsnachfrage auf die Schweiz. In *paper presented at Heureka '08*, Stuttgart, Germany, March 2008.

[27] Y. Murakami, T. Ishida, T. Kawasoe, and R. Hishiyama. Scenario description for multi-agent

simulation. In *Autonomous agents and multiagent systems (AAMAS'03)*, Melbourne, Australia, July 2003.

[28] K. Nagel and M. Schreckenberg. A cellular automaton model for freeway traffic. *Journal de Physique I France*, 2:2221–2229, 1992.

[29] K. Nishinari, A. Kirchner, A. Nazami, and A. Schadschneider. Extended floor field CA model for evacuation dynamics. In *Special Issue on Cellular Automata of IEICE Transactions on Information and Systems*, volume E84-D, January 2001.

[30] X. Pan, C. Han, K. Dauber, and K. Law. A multi-agent based framework for the simulation of human and social behavior during emergency evacuations. *AI and Society*, 22(2):113–132, 2007.

[31] W. Predtetschenski and A. Milinski. *Planning for Foot Traffic in Buildings*. Amerind Publishing Co. Pvt. Ltd., New Delhi, 1978.

[32] B. Raney, N. Cetin, A. Völlmy, M. Vrtic, K. Axhausen, and K. Nagel. An agent-based microsimulation model of Swiss travel: First results. *Networks and Spatial Economics*, 3(1):23–41, 2003.

[33] B. Raney and K. Nagel. An improved framework for large-scale multi-agent simulations of travel behavior. In P. Rietveld, B. Jourquin, and K. Westin, editors, *Towards better performing European Transportation Systems*. Routledge, London, 2006. Similar version TRB preprint number 05-1846.

[34] A. Schadschneider, W. Klingsch, H. Klüpfel, T. Kretz, C. Rogsch, and A. Seyfried. Evacuation dynamics: Empirical results, modelling and applications. In B. Meyers, editor, *Encyclopedia of Complexity and System Science*. Springer, Berlin, to appear.

[35] M. Schreckenberg and S. D. Sharma, editors. *Pedestrian and Evacation Dynamics*. Proceedings of the 1st international conference, Duisburg, 2001. Springer, 2001.

[36] P. M. Simon, J. Esser, and K. Nagel. Simple queueing model applied to the city of Portland. *International Journal of Modern Physics C*, 10(5):941–960, 1999.

[37] U. Weidmann. *Transporttechnik der Fussgänger*, volume 90 of *Schriftenreihe des IVT*. Institute for Transport Planning and Systems ETH Zürich, 2 edition, 1993. in German.