EVACUATION SIMULATION WITH LIMITED CAPACITY SINKS

An evolutionary approach to solve the shelter allocation and capacity problem in a multi-agent evacuation simulation

Gunnar Flötteröd
Transport and Mobility Laboratory, EPFL - Ecole Polytechnique Fdrale de Lausanne, Lausanne, Switzerland
gunnar.floetteroed@epfl.ch

Gregor Lämmel
Transport Systems Planning and Transport Telematics, Berlin Institute of Technology, Berlin, Germany
laemmel@vsp.tu-berlin.de

Keywords: simulated annealing, iterative learning, Nash equilibrium, system optimum.

Abstract: Evacuation modeling is an important problem that has been intensively studied in the last 40 years. In a general evacuation situation, there might be more then one safe place. A common approach to reduce this multi-destination problem to a single destination problem is to connect all safe places via zero cost links to a super sink. However, this approach works only as long as the safe places have no capacity constraints. In this paper, we present an approach to solve an evacuation problem where the safe places have limited capacities. This is, for example, the case if the safe places are shelter buildings for tsunami evacuation. The problem is solved through an evolutionary learning algorithm for the combined route- and shelter-assignment problem combined with a heuristic approach for the fair minimization of shelter capacities. Different behavioral assumptions ("fair" vs. "globally optimal") are investigated. The proposed approaches are discussed in the context of a real-world tsunami evacuation problem.

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. Important tools for the planning of organized reactions to such events are model-based simulation systems.

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., (Sheffi, 1985), and dynamic approaches, e.g., (Peeta and Ziliaskopoulos, 2001).

A typical static evacuation simulation is MASSVAC (Hobeika and Kim, 1998). 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 adopted to evacuation scenarios, e.g., MITSIM (Jha et al., 2004), DYNASMART (Kwon and Pitt, 2005), and PARAMICS (Chen and Zhan, 2004).

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 above DTA packages rely on microscopic traffic models. Further microscopic approaches that have been applied to the simulation of evacuation dynamics are cellular automata (Klüpfel et al., 2003) and the social force model (Helbing et al., 2002). Examples of software packages based on macroscopic models are ASERI (Schneider and Könnecke, 2002) and Simulex (www.iesve.com).

Random utility models are also applicable to the microscopic modeling of pedestrian dynamics, however, they are yet to be applied in evacuation scenarios (Bierlaire et al., 2003).

The general evacuation problem is to minimize the egress time of an endangered region or building by assigning a feasible escape route and destination to every evacuee. This problem is complex because of congestion effects that inevitably occur when many evacuees enter the transportation facilities (roads, hallways, stairways) at once.

In some evacuation scenarios there secure areas with with limited capacity available within the evacuation zone, such as shelter buildings in tsunami prone areas. An evacuation that accounts for shelters bears similarities with the capacitated warehouse location problem (CWLP), e.g., (Akinc and Khumawala, 1977). Given a constellation of warehouses, the CWLP is to satisfy the demand for goods of a number of customers in a way that minizes the total transportation costs without exceeding the warehouses capacity. A concrete application of mathematical programming techniques to the evacuee–shelter–allocation problem is (Sherali et al., 1991).

This paper deploys a detailed microsimulation for the representation, analysis, and optimization of pedestrian evacuation dynamics for a tsunami situation in a large coastal metropolitan area. Building on existing routing strategies (Removed, XXXX), it provides new solutions to (i) the combined route and shelter assignment problem and (ii) the shelter capacity optimization problem, considering both "fair" and "optimal" assignment rules.

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). This is an advantage over macroscopic models in that it allows (at least technically) for a much higher model resolution. However, it comes at the price of greater difficulties in the mathematical treatment of the problem. The solution presented in this article are therefore only of an approximate nature.

The remainder of this article is organized as follows. Section 2 gives a detailed description of the considered evacuation problem. After revisiting the pure evacuation route assignment problem in Sections 3, the combined route and shelter assignment problem is presented in Section 4, followed by the shelter capacity optimization problem in Section 5. Detailed simulation results are given next in Section 6. A final discussion of the results is given together with a summary of the findings in Section 7.

2 PROBLEM STATEMENT

We investigate different strategies to assign routes and destinations (shelters) to evacuees. In a second step, we identify optimal dimensions of the shelters. Overall, we consider two different objectives:

Fairness

: No evacuee will agree to take an obvious detour when heading for a shelter or to select an obviously faraway shelter instead of a nearby one. This requires to identify route and shelter assignments that are fair in that no evacuee can obviously gain by switching to a different route or shelter. This calls for a route and shelter assignment that results in a Nash equilibrium of all evacuation strategies in the population.

Efficiency

: It is desirable to evacuate the system as quickly as possible, which is equivalent to minimizing the total evacuation time of the whole population (see below). While a Nash strategy has the obvious and important advantage of general acceptance, it may be suboptimal in this regard because some evacuees may do great damage to others by blocking their ways/shelters. In this research, we identify approximations of optimal evacuation strategies as benchmarks to which fair solutions can be compared.

An important topic for future research is to combine efficient and fair solution strategies into evacuation plans that are more efficient than pure Nash equilibria but without introducing obvious levels of unfairness.

2.1 Simulation framework

We model the urban evacuation region and the population of evacuees with a multi-agent simulation, where every single person is individually represented. For this purpose, the MATSim (MATSim, 2010) simulation framework is adopted. MATSim is designed for the computation of transport equilibria, and hence it can be immediately deployed for the computation of Nash evacuation strategies. For approximately optimal strategies, however, some adjustments are necessary.

MATSim allows for adjustments in the different choice dimensions of a simulated traveler through *modules*, where, typically, one module is responsible for one particular choice dimension. In our application, this requires to specify four modules: (1) Nash route choice, (2) Nash destination (shelter) choice, (3) optimal route choice, (4) optimal shelter choice.

MATSim computes approximate Nash equilibria by iterating best-response behavior: in every iteration, a fraction of the travelers re-calculates a route or a destination based on what would have been best in the previous iteration, assuming that the behavior of all other agents stays unchanged. After this replanning, the resulting *plans* of all travelers are simultaneously executed in the mobility simulation and new performance measures are computed. This process is repeated many times. If it stabilizes, then a route or

destination re-planning does not result in a substantial improvement any more, and an approximate Nash equilibrium is obtained.

An alternative assignment logic is to not compute best responses in every iteration but *cooperative* behaviors that improve the situation of the population as a whole. The concrete realization of such behavioral patterns within the MATSim framework is more involved and is hence postponed to later parts of this article.

2.2 Network modeling

We model the evacuation network as a directed graph consisting of a node set $\mathcal{N}$, a link set $\mathcal{L} \subset \mathcal{N} \times \mathcal{N}$, a set of source (origin) nodes $\mathcal{S} \subset \mathcal{N}$ and a set of sink (destination, shelter) nodes $\mathcal{D} \subset \mathcal{N}$.

Every destination $d \in \mathcal{D}$ has a capacity c_d that represents the maximum number of evacuees that can be sheltered at this destination. Destination nodes may also be located at the boundary of the endangered area, in which case they to not provide a limited shelter but access to a safe region, which is modeled by assigning them an unlimited shelter capacity.

We consider a pedestrian simulation scenario on a road network, where the crossings are modeled as nodes and the street segments connecting the crossings are modeled as links. The physics of pedestrian traffic flow dynamics are modeled through parameters of the link. The most important parameter is the flow capacity, which describes how many flow units can travel along a link per time unit. Another important parameter is the time-dependent link travel time, which describes the expected travel time when entering a link at a certain point of time. Note that this formalism can be immediately transferred to vehicular evacuation problems.

For illustration, an evacuation network with bounded-capacity shelters is given in figure 1.

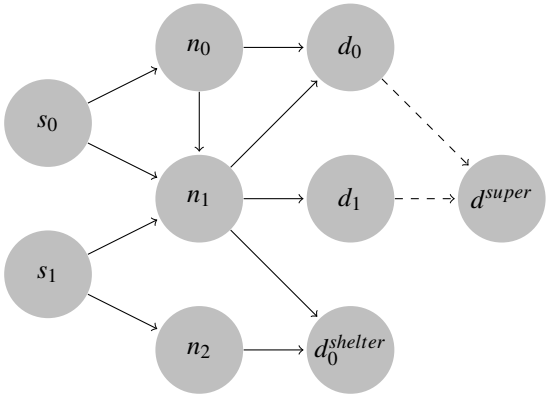

Figure 1: Example evacuation network with shelter nodes

3 ROUTE ASSIGNMENT

Given that every evacuee $n = 1 \ldots N$ is assigned to a shelter $d(n)$, the route assignment problem is to find a feasible and in some sense best route from that evacuee's origin $s(n)$ to her shelter.

3.1 Shortest path solution

The most straightforward approach to an evacuation problem is to compute shortest paths based on some pre-defined criterion such as distance or average travel time. While this approach is computationally efficient (Dijkstra, 1959) and straightforward to implement in a microscopic evacuation simulation, it is of limited practical use.

The main problem is that the shortest path solution does not take congestion into account. The time it takes to travel through a link (the link travel time) depends on the congestion level on that link: the flow capacity of a link defines how many flow units can leave it per time interval. If there are too many evacuees on the link, they queue up and have to wait.

Since the shortest path solution computes evacuation routes *a priori*, it does not account for time-dependent congestion effects, which renders it a poor evacuation strategy. The following two subsections present improved routing solutions.

3.2 Nash equilibrium approach

In the given context, a Nash equilibrium describes a situation where no evacuee can gain by unilaterally deviating from her current (routing) strategy (Nash, 1951). In most evacuation situations, the Nash equilibrium leads to a shorter overall evacuation time than the naive shortest path solution. Furthermore, since a Nash equilibrium means that nobody has an incentive to deviate, it can be considered as a socially acceptable and hence implementable evacuation strategy.

In a multi-agent evacuation simulation the solution can be moved towards a Nash equilibrium through iterative learning (Gawron, 1998; Removed, XXXX). As described above, an iterative learning algorithm starts with a given (routing) strategy pattern and then adjusts it through trial and error efforts of the simulated agents. In the given evacuation context, strategies are only evaluated based on their travel time.

Formally, the real-valued time is discretized into

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 $C_a(k) = \tau_a(k)$
   (b) simulate agent movements, obtain new $\tau_a(k)$ for all a and k

Algorithm 2 System optimum approach

1. initialize $C^s_a(k) = 0$ and $\tau_a(k)$ 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 $C_a(k) = \tau_a(k) + C^s_a(k)$
   (b) load vehicles on network, obtain new $\tau_a(k)$ and $C^s_a(k)$ for all a and k

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.

3.3 Approximately system optimal assignment

A system optimal routing solution minimizes the total travel time in the system. Classical solutions to this problem apply mathematical programming techniques, which are based on the theory of dynamic network flows. The foundations of these techniques have been laid in (Ford and Fulkerson, 1962), and dynamic flow models have been applied to evacuation problems from the early 1980s on (see, e.g., (Chalmet et al., 1982)).

The mathematical theory that underlies the above methodology provides an important result regarding the notion of "optimality", which puts the following problems into perspective:

- Minimization of the egress time.
- Minimization of the average (or total) travel time.
- Maximization of the amount of flow that has already reached the sink at each time step.

The triple optimization theorem proved by Jarvis and Ratliff (Jarvis and Ratliff, 1982) states that the solution which minimizes average evacuation time also maximizes the amount of flow that has reached the sink at each time step and therefore also minimizes the egress time.

In the multi-agent domain, an approximate system optimum can be found through an iterative learning approach that is closely related to the simulation of a Nash equilibrium as described above (Removed, XXXX). The procedure is called Approximate System Optimal Assignment (SO).

The only difference to the Nash routing logic is that the travel time based on which agents evaluate their routes is replaced by the marginal travel time (Peeta and Mahmassani, 1995). The marginal travel time of a link is the amount by which the total system travel time changes if one additional vehicle drives along that link. It is the sum of the cost experienced by the added vehicle ($\tau_a(k)$) and the cost imposed on other vehicles. The latter is denoted here as the time-dependent "social cost" $C^s_a(k)$.

Letting each agent *individually* minimize its marginal travel time implicitly enforces a *cooperative* behavior that also minimizes the total system travel time. It therefore also maximizes the amount of evacuees that have already reached the sink at each time step, which also minimizes the egress time.

Alg. 2 outlines the arguably most straightforward implementation of this approach in a time-discrete multi-agent simulation.

4 SHELTER ASSIGNMENT

The shelter assignment problem is to identify, for each evacuee, if this evacuee should access a shelter or not and, given that a shelter is accessed, to decide which shelter. Again, both a Nash and an SO approach are possible.

In either case, the (re-)allocation of an agent to a shelter requires also to re-compute its route. We maintain consistency here in that the Nash-shelter assignment is combined with a Nash-route assignment and the SO-shelter assignment is combined with an SO-route assignment

4.1 Nash shelter assignment

Due to the given capacity constraints of all shelters, a naive best-response simulation of a Nash shelter assignment is infeasible: assuming that all shelter capacity is utilized in an evacuation situation, the best response of an agent having managed to get into a shelter is to stick to this strategy forever.

We therefore extend the best-response simulation logic to a pair-wise best response, where for every

Algorithm 3 Heuristic Nash shelter allocation

1. initialize routes and destinations for all agents
2. repeat many times
   (a) load all agents on the network
   (b) extract link travel times and social costs
   (c) for every agent $n = 1 \ldots N$, do with P_{replan}:
   - with $P_{reroute}$, compute a new route from $s(n)$ to $d(n)$.
   - with P_{move},
     i. randomly select a non-full shelter d'
     ii. compute the benefit of a move: $\delta = c_{s(n)d(n)} - c_{s(n)d'}$
     iii. if $\delta > 0$, assign d' as the new destination to n and re-route n
   - with P_{switch},
     i. randomly select n' from $\{1, \ldots, N\}$
     ii. compute the minimum benefit of a switch: $\delta = min(c_{s(n)d(n)} - c_{s(n)d(n')}, c_{s(n')d(n')} - c_{s(n')d(n)})$
     iii. if $\delta > 0$, then switch the destinations of n and n' and re-route both agents

Algorithm 4 Heuristic SO shelter allocation

1. initialize routes and destinations for all agents
2. repeat many times
   (a) load all agents on the network
   (b) extract link travel times and social costs
   (c) for every agent $n = 1 \ldots N$, do with P_{replan}:
   - with $P_{reroute}$, compute a new route from $s(n)$ to $d(n)$.
   - with P_{move},
     i. randomly select a non-full shelter d'
     ii. compute the benefit of a move: $\delta = c_{s(n)d(n)} - c_{s(n)d'}$
     iii. if $\delta > 0$, assign d' as the new destination to n and re-route n
   - with P_{switch},
     i. randomly select n' from $\{1, \ldots, N\}$
     ii. compute the cost change of a switch: $\delta = c_{s(n)d(n)} + c_{s(n')d(n')} - c_{s(n)d(n')} - c_{s(n')d(n)}$
     iii. if $delta > 0$, then switch the destinations of n and n' and re-route both agents

agent n that re-plans its shelter assignment a "switching partner" n' in another shelter is randomly selected, and both agents switch their shelters if and only if *both* benefit from this switch. This decision is made based on the expected travel time of a best-response re-routing to the new destination, which also is adopted in the case of an accomplished switch.

The iterative simulation conducts a shelter switch with a certain probability P_{switch}, and it also maintains the option of a plain route recomputation with $P_{reroute}$. Algorithm 3 defines the details of this logic.

Step 1., 2.(a), 2.(b) and the first step of 2.(c) are symmetric to the routing logic of Algorithm 1. In the shelter allocation part of the algorithm, an agent moves with probability $P_{replan} * P_{move}$ to a non-full shelter if she would benefit from that move. With probability $P_{replan} * P_{switch}$, two agents switch their shelters if both of them would benefit. The origin of agent n is denoted by $s(n)$ and the destination by $d(n)$. The cost $c_{s(n)d(n)}$ for agent n corresponds to the travel time for agent n from $s(n)$ to $d(n)$.

4.2 SO shelter assignment

Technically, the SO shelter assignment does not function differently from the Nash shelter assignment, only that two agents now "agree" to switch their shelters if this reduces the total travel time in the system.

To decide this, the expected change in *marginal* travel times is evaluated before and after the switch. Recall that marginal cost is defined as the sum of individual cost and social cost. The change in individual cost (travel time) captures the (dis)benefits only of the two agents negotiating the switch, whereas the additional change in social cost (travel time) captures the effect of this switch on all other evacuees.

Apart from the different cost evaluations, the simulation logic stays unchanged. Algorithm 4 gives the details.

Step 1., 2.(a), 2.(b) and the first step of 2.(c) are symmetric to the routing logic of Algorithm 2. In the shelter allocation part of the algorithm, an agent moves with probability $P_{replan} * P_{move}$ to a non-full shelter if she would benefit from that move. With probability $P_{replan} * P_{switch}$, two agents switch their shelters if this would decrease the total system travel time. The origin of agent n is denoted by $s(n)$ and the destination by $d(n)$. The cost $c_{s(n)d(n)}$ for agent n corresponds to agent n's marginal travel time as explained in Section 3.3.

5 SHELTER CAPACITY ASSIGNMENT

The shelter capacity assignment problem is to minimize the total shelter capacity over the entire evacuation area subject to the constraint that no evacuee

takes damage from being neither able to reach the safe area nor to enter a shelter because of lacking capacity.

That is, a shelter capacity configuration is required where only those evacuees are assigned to shelters who would not make it to the safe region otherwise and where the shelter capacities are tailored towards these evacuees only. The main difficulty of this problem is owed to the simulation-based representation of "needing a shelter".

The previous section demonstrates how a combined route choice and shelter allocation Nash equilibrium and SO assignment can be simulated. In this section, we enhance this assignment by a shelter capacity adjustment logic. The main difficulty to be dealt with here is that two coupled assignment problems need to be simulated at once: one for the evacuees who need the shelters, including the choice dimension of accessing a shelter; and one for those evacuees who do not need the shelters, excluding them from the option of accessing a shelter.

5.1 Shelter capacity assignment subject to double Nash constraints

The iterative simulation logic that allows for both re-routing and shelter switching is maintained. In order to obtain a Nash route choice pattern, the best-response re-routing logic is adopted.

If there is more shelter capacity than strictly needed, there will also be agents in the shelters that do not need the shelter (because it can be assumed that for many such agents the shelter still is closer than the safe area). It is not feasible to *ex post* remove these agents from the shelters and to constrain the shelter capacities accordingly because this would change the travel times and hence the survival chances of the needy agents. The shelter capacities therefore need to be gradually changed during the iterations.

This effect is achieved by evaluating, in every iteration, the space occupied in every single shelter by agents that would also have made it to the safe region. If this surplus is vanishing, the shelter is urgently needed and its capacity is increased by a relative amount (say, 5 percent). If, on the other hand, this surplus is substantial, the shelter is too large, and it is shrunk by a relative amount (say, again, 5 percent) of its surplus capacity.

This mechanism, in combination with the strict preference for needy agents in the shelter allocation, eventually leads to a configuration where all available shelter capacity is allocated to needy agents, given otherwise fair Nash equilibrium conditions. Algorithm 5 gives an overview.

The term x_{nd} indicates the allocation of agent n to shelter d, i.e., $x_{nd} = 1 \Leftrightarrow d(n) = d$. $P_{capacity}$ denotes the relative amount by which the capacity of a shelter can change at most. The super shelter d^{super} is a shelter with unbounded capacity that represents the entire safe area as depicted in Figure 1.

Algorithm 5 Heuristic Nash shelter allocation and capacity

1. initialize routes and destinations for all agents
2. repeat many times
   (a) load all agents on the network
   (b) extract link travel times and social costs
   (c) for every agent $n = 1 \ldots N$, do with P_{replan} Nash re-planning
   (d) for every shelter $d = 1 \ldots D$, do:
   - $o(d) = c(d) - \sum_{n=1}^{N} x_{nd}$
   - $\delta_d = 0$
   - for every agent $n = 1 \ldots N$ with $x_{nd} = 1$, do:
     i. calculate $\delta_{s(n)d^{super}}$ the surplus of time that n has to reach d^{super} just in time
     ii. $\delta_d = \delta_d + \delta_{s(n)d^{super}}$
     iii. if $\delta_{s(n)d^{super}} > 0$, then increase $o(d)$ by one
   - if $o(d) > 0$, then:
   - decrease $c(d)$ by $min(o(d), P_{capacity} * c(d))$; re-route agents with $\delta_{s(n)d^{super}} > 0$ to d^{super} if needed
   - else: increase $c(d)$ by $P_{capacity} * c(d))$

5.2 Shelter capacity assignment subject to SO constraints

The only change when going from a shelter capacity assignment subject to Nash constraints to one subject to SO constraints is that the route choice and shelter switching behavior of all re-planning agents is conducted according to the SO logic described in Subsections 3.3 and 4.2. For completeness, Algorithm 6 gives an overview.

The conditions for shelter capacity decreases and increases in the SO case are the same as for the Nash case given in in Algorithm 5.

6 EXPERIMENTS

The different route and shelter assignment models are tested in combination with the capacity optimization logic on a real world scenario for the Indonesian city of Padang. Padang is located at the West Coast of

Algorithm 6 Heuristic SO shelter allocation and capacity

1. initialize routes and destinations for all agents
2. repeat many times
   (a) load all agents on the network
   (b) extract link travel times and social costs
   (c) for every agent $n = 1 \ldots N$, do with P_{replan} SO re-planning
   (d) for every shelter $d = 1 \ldots D$, do:
   - $o(d) = c(d) - \sum_{n=1}^{N} x_{nd}$
   - $\delta_d = 0$
   - for every agent $n = 1 \ldots N$ with $x_{nd} = 1$, do:
     i. calculate $\delta_{s(n)d^{super}}$ the surplus of time that n has to reach d^{super} just in time
     ii. $\delta_d = \delta_d + \delta_{s(n)d^{super}}$
     iii. if $\delta_{s(n)d^{super}} > 0$, then increase $o(d)$ by one
   - if $o(d) > 0$, then: decrease $c(d)$ by $min(o(d), P_{capacity} * c(d))$; re-route agents with $\delta_{s(n)d^{super}} > 0$ to d^{super} if needed
   - else: increase $c(d)$ by $P_{capacity} * c(d))$

Sumatra Island. It is exposed to earth quake triggered tsunamis.

The evacuation street network consist of approx. 12 500 unidirectional links and 4 500 nodes. In ongoing surveys, the buildings of the city are investigated if they could serve as shelters during tsunami evacuations. For this work, 42 hypothetical shelter buildings with a total capacity of roughly 31 500 evacuees are placed in the network. A sketch of the evacuation network including the shelter buildings is given in Figure 2. The gray-shaded area has to be evacuated. This area corresponds to what is assumed to be flooded by a tsunami plus an additional spatial buffer of 500 m. The shelter buildings are colored in green.

There is in total a number of 224 798 evacuees. This corresponds to the number of person living within the evacuation area. Since the shelter capacity is only about 31 500, not all evacuees can find a safe place within the evacuation area. Those evacuees have to leave the evacuation area in order to be safe.

To compare the performance of the different proposed approaches, we conducted four different simulations.

- *Run 1* implements the (approx.) Nash equilibrium routing logic with Nash shelter allocation.
- *Run 2* implements the (approx.) Nash equilibrium routing logic with Nash shelter allocation and shelter capacity assignment.
- *Run 3* implements the (approx.) system optimum routing logic with SO shelter allocation.
- *Run 4* implements the (approx.) system optimum routing logic with SO shelter allocation and shelter capacity assignment.

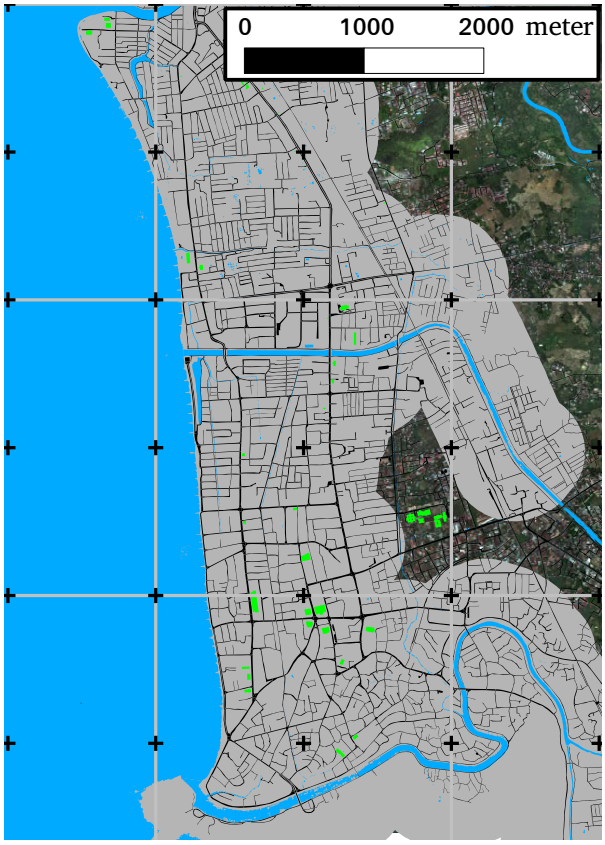

Figure 2: Map of the evacuation area.

These runs are performed with a 10% sample of the population. The shelter capacities and network flow parameters are accordingly scaled down to 10%. This procedure saves computing time while staying reasonably realistic. With this setup, a simulation with 2000 learning iterations takes between 05:30 h (*Run 1*) and 10:30 h (*Run 4*) on a 2.66 GHz CPU running 64 bit Java on Linux. The maximum memory consumption is less than 3 GB in all experiments.

At the beginning of all runs the agents are assigned randomly to the shelters. Figure 3 shows the average evacuation time per agent over the iteration number. The curve of *Run 1* clearly shows that the Nash shelter allocation assignment algorithm leads to considerable better evacuation times compared to a random agent shelter assignment. With the addition of the shelter capacity assignment algorithm the average evacuation time can further be reduced (*Run 2*). How-

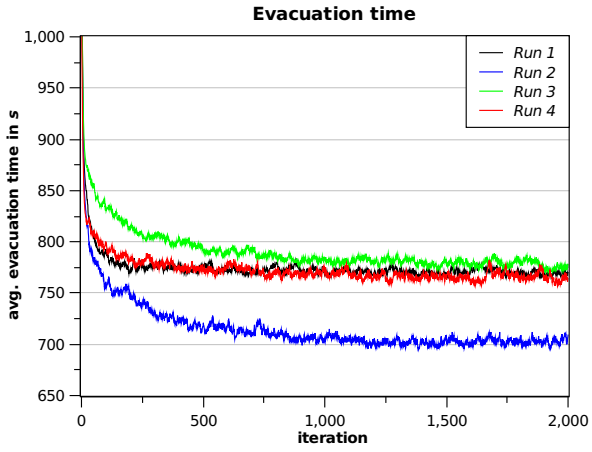

Figure 3: Average evacuation time per agent versus iteration number.

ever, the evacuation time in *Run 2* can only be reduced by means of an increase in shelter capacities, which is shown in Figure 4. Since the shelter capacity assignment algorithm accepts only those agents to be in a shelter for whom the super shelter (area outside the evacuation zone) is not reachable in time, the resulting shelter capacities in *Run 2* are minimal for a Nash solution of the evacuation problem.

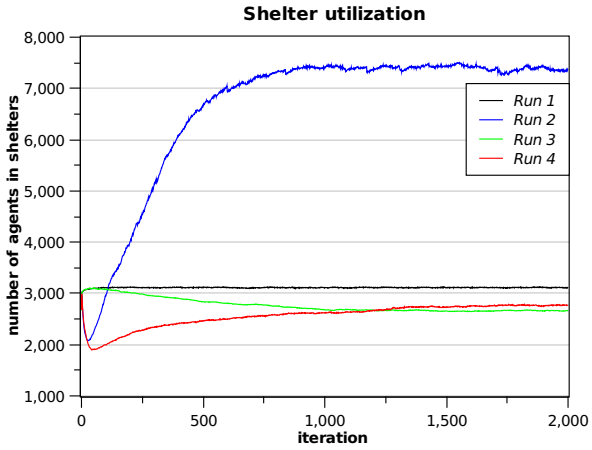

Figure 4: Change of the shelters total utilization over the learning iterations.

The SO in *Run 3* and *Run 4* is realized by adding social costs to the travel time on each link. However, the resulting behavior is not a consequence of an intrinsic motivation but would have to be enforced externally. Therefore, these runs should be considered as benchmark solutions.

Comparing the average evacuation time of *Run 1*, *Run 3* in figure 3 and those of *Run 2*, *Run 4* respectively, it is to notice that the average evacuation time is not better in the SO case than in the Nash case. For *Run 4* the average evacuation time is even worse than in *Run 2*. However, the results are not absolutely comparable since in *Run 3* much more agents survive compared to *Run 1*. This is shown in figure 5. In the Nash case (*Run 1*), more than 100 agents do not mange to escape even after 2000 iterations of learning, whereas in the SO case (*Run 3*) about 20 agents do not have enough time. The average evacuation time in *Run 2* (Nash with shelter allocation and capacity change) is about 1 minute less compared to the other runs. However, the gain by one minute could only be achieved by an increase of the shelters capacity by more than 400% (see figure 4).

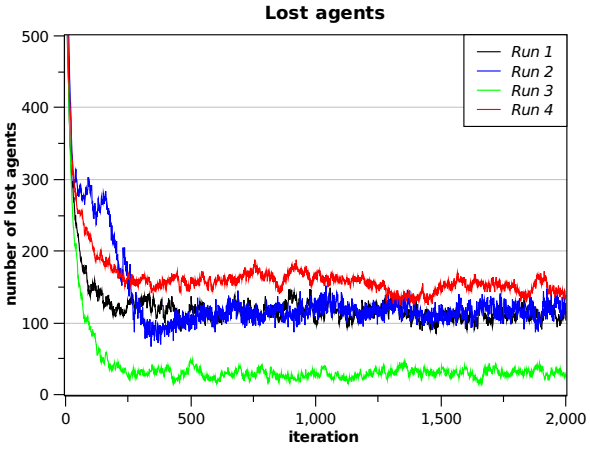

Figure 5: Number of agents that would need more time to evacuation versus iteration number.

At the same time, the shelters capacity in *Run 4* declines by achieving a comparable evacuation times to *Run 1* and *Run 3*. Even in *Run 3* where the shelters capacity is not explicitly changed the shelters are not fully utilized. To sum up, in the Nash equilibrium more shelter capacity is needed compared to the System optimum. A reason for this phenomenon might be that the SO routing reduces the congestion in the network such that less agents are in need of a shelter. This would also explain why more agents survive in the SO scenarios.

7 DISCUSSION AND SUMMARY

The best strategy for a tsunami-threatened city like Padang would arguably be to build a tsunami proof

shelter for every person. However, this would exceed any funding resources. The relevant question here is how many shelters with which capacity are needed to obtain a feasible solution where everyone is safe who has no chance otherwise.

The proposed algorithm help to determine these numbers. Furthermore, the algorithm for the shelter capacity assignment can also be adapted to figure good locations for shelter buildings. For this, one would have to start with a very high number of randomly distribute shelter buildings and then from time to time remove underutilized shelters until the number of shelter buildings is reduced to the desired amount.

The main difference in the simulation results is that the needed shelter capacity in the Nash equilibrium case is much higher than in the SO case. If one wants to achieve the highest benefits with the least effort, than one could implement the shelter configuration of *Run 4* and distribute some kind of tickets to the people that are allowed to enter a shelter. To stay fair, those tickets could for example handed out preferred to the most vulnerable people like elderly people or pregnant women.

Summarizing, a learning framework to solve the shelter allocation and capacity optimization problem is introduced and tested on a real world scenario. The learning framework can configured either to get an approximately Nash equilibrium where the individual travel times are minimized or an approximately system optimum where the system travel time is minimized. Results for a real world scenario show that both approaches give feasible results. However, in the Nash approach more shelter capacity is needed compared to the system optimum. It seems to be that the Nash equilibrium produces more congestion compared to the system optimum and therefore more agent are in need for shelter since their travel times would be to long to reach the safe area.

An interesting topic for further research is to investigate the relative effect of capacity improvements in the transportation system when compared to investments in increased shelter capacities.

ACKNOWLEDGEMENTS

Removed for blind review.

REFERENCES

Akinc, U. and Khumawala, B. (1977). An efficient branch and bound algorithm for the capacitated warehouse location problem. *Management Science*, 23(6):585 – 594.

Bierlaire, M., Antonini, G., and Weber, M. (2003). Behavioral dynamics for pedestrians. In Axhausen, K., editor, *Moving through nets: The physical and social dimensions of travel*. Elsevier.

Chalmet, L., Francis, R., and Saunders, P. (1982). Network models for building evacuation. *Management Science*, 28:86–105.

Chen, X. and Zhan, F. (2004). 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.

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

Ford, L. and Fulkerson, D. (1962). *Flows in Networks*. Princeton University Press.

Gawron, C. (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.

Helbing, D., Farkas, I., Molnar, P., and Vicsek, T. (2002). Simulation of pedestrian crowds in normal and evacuation situations. In (Schreckenberg and Sharma, 2002), pages 21–58.

Hobeika, A. and Kim, C. (1998). Comparison of traffic assignments in evacuation modeling. *IEEE Transactions on Engineering Management*, 45(2):192–198.

Jarvis, J. and Ratliff, H. (1982). Some equivalent objectives for dynamic network flow problems. *Management Science*, 28:106–108.

Jha, M., Moore, K., and Pashaie, B. (2004). Emergency evacuation planning with microscopic traffic simulation. Paper 04-2414, Transportation Research Board Annual Meeting, Washington, D.C.

Klüpfel, H., Meyer-König, T., Keßel, A., and Schreckenberg, M. (2003). Simulating evacuation processes and comparison to empirical results. In Fukui et al, M., editor, *Traffic and granular flow '01*, pages 449–454. Springer, Berlin Heidelberg New York.

Kwon, E. and Pitt, S. (2005). 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.

MATSim (accessed 2010). MATSim web site. http://www.matsim.org.

Nash, J. (1951). Non-cooperative games. *Annals of Mathematics*, 54(2):286–295.

Peeta, S. and Mahmassani, H. (1995). System optimal and user equilibrium time-dependent traffic assignment in congested networks. *Annals of Operations Research*, 60:81–113.

Peeta, S. and Ziliaskopoulos, A. (2001). Foundations of Dynamic Traffic Assignment: The Past, the Present and the Future. *Networks and Spatial Economics*, 1(3):233–265.

Removed (XXXX). Placeholder for own work (blind review).

Schneider, V. and Könnecke, R. (2002). Simulating evacuation processes with ASERI. In (Schreckenberg and Sharma, 2002), pages 303–314.

Schreckenberg, M. and Sharma, S. D., editors (2002). *Pedestrian and Evacation Dynamics*. Proceedings of the 1st international conference, Duisburg, 2001. Springer.

Sheffi, Y. (1985). *Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods*. Prentice-Hall, Englewood Cliffs, NJ, USA.

Sherali, H., Carter, T., and Hobeika, A. (1991). A location-allocation model and algorithm for evacuation planning under hurricane/flood conditions. *Transportation Research Part B: Methodological*, 25(6):439 – 452.