Dynamic Multi-Objective Evacuation Path Planning in Mobile Ad Hoc Networks

Technical Report

Sabrina Merkel*, Alexander Kesseler† and Hartmut Schmeck‡

Karlsruhe Institute of Technology

January 21, 2014

Abstract

We propose an approach to compute navigation instructions on mobile devices carried by people during a building evacuation in order to guide them to safe areas or exits. The mobile devices form an ad hoc network via local communication links and use this network to collect information about the current evacuation situation. This information is used for path planning in order to optimize escape routes with respect to multiple objectives, such as congestion avoidance and risk minimization. Due to delays and link breakages in the network communication, the prediction of emerging congestions becomes a major challenge. We propose two congestion indicators which are based on uncertain knowledge gained from local communication between the mobile devices. It is shown that dynamic multi-objective evacuation path planning reduces congestions and accelerates the evacuation process compared to a state-of-the-art evacuation planning approach for mobile devices.

1 Introduction

The growing world population and the simultaneous technological progress let buildings become increasingly large and more complex. Hence, the question of how to evacuate people from such buildings as fast and smoothly as possible becomes all the more important. It is urgent to address this matter, especially, since panic often causes deaths and injuries. Nevertheless, the equipment of buildings for evacuation support is comparatively outdated. Todays approaches

*sabrina.merkel@kit.edu

†alexander.kesseler@web.de

‡hartmut.schmeck@kti.edu

to prevent disasters and support a well-ordered evacuation are to install safety devices, such as sprinkler systems, fire or smoke alarms, fire extinguishing installations, emergency exits, fire escapes, exit signs, and emergency maps. In [6], reports from survivors of the attacks of September 11th, 2001 indicate that the main obstacles during escape are congestions in bottleneck-areas and closed or blocked passages due to smoke formation or damaged building fabrics. In addition, it is reported that even people who were familiar with the building's layout had difficulties to find their way due to smoke formation or lack of illumination. Although some of these problems can be addressed with appropriate signage or emergency lights, there remains the open question about what to do when a designated escape route is not usable anymore. Exit signs and emergency maps in todays buildings are static and analogous. Consequently, they are not able to adapt to unforeseen changes.
The rapidly growing spread of communication devices, such as smart phones or tablet PCs, opens up new opportunities to meet the challenges of a modern, adaptable evacuation system. The mobile devices can be used to point out escape directions and navigate their users to safe areas or exits. Since these devices possess processing capabilities, they can process information about changes in the evacuation environment and adapt the path planning accordingly. The ability of mobile devices to form ad hoc networks through wireless communication with other nearby devices offers the possibility to exchange information about the current evacuation situation. This knowledge can be integrated in evacuation route planning to decide for the optimal way leading outside the building under the given circumstances. Apart from the given adaptability, mobile devices in an ad hoc network offer the advantage that there is no need for a central computing unit which would be a potential single-point of failure. The failure of a central server could lead to a total breakdown of the entire evacuation support, which is why decentralized systems are especially attractive for deployment in emergency situations.
The circumvention of potential congestions is an obvious objective when choosing an evacuation route. But, apart from that, there are other criteria to consider. For example, the avoidance of risky areas can be crucial for the safety of the evacuees. Additionally, each criterion can be more important for some evacuees than others. A handicapped person, for example, would probably be willing to take a longer path if it is more accessible. Therefore, we propose the dynamic multi-objective distributed evacuation planning (DMO-DEP) which is capable to consider many aspects simultaneously and to weight them depending on the evacuees' individual characteristics. The hereafter addressed objectives are risk minimization and congestion avoidance, although it is easily possible to further extend the list of considered criteria. To able to adapt the path planning to newly available information we propose to use an incremental, heuristic search algorithm from the area of robotics, called D* Lite [11] to compute the optimal path for each evacuee to an exit of the building. D* Lite includes previous search results when adapting to changes and, hence, reduces necessary computations. Experiments show that the DMO-DEP approach is able to improve the overall evacuation time by decentralized path planning on mobile devices based on

local information compared to a situation without communication between the devices. Additionally, the evacuation process can be accelerated and waiting times for evacuees are reduced compared to a state-of-the-art distributed path planning algorithm for mobile devices.
This paper is structured as follows. Section 2 gives an overview of related work. The D* Lite algorithm is described in Section 3. Section 4 introduces the DMO-DEP algorithm and Section 5 contains an experimental investigation of the performance of DMO-DEP. Section 6 concludes the paper.

2 Related Work

Route optimization, or path planning, is usually performed on a macroscopic graph representation of the environment, where nodes represent different regions in which people are gathered (e.g. rooms) and edges link nodes which are connected by doors or hallways in the real building. Such a graph model serves as a basis to calculate the quickest flow, i.e. minimizing the time for a certain number of evacuees to reach the exit. Exact solutions to this problem can be found with a time-expansion of the underlying graph model, where multiple copies for each node are made to represent the node's state at a certain time step [1]. In [12], a heuristic approach to solve the quickest flow problem is presented. It adapts a routing protocol to stepwise schedule evacuees on various paths by reserving capacities on the nodes and edges for the time when they are occupied. Another routing protocol is adapted in [18] to achieve load balanced traveling of pedestrians in a building between fixed destinations. In evacuation planning, however, the goal is to find the optimal path for a single evacuee instead of optimizing flows, i.e. sets of evacuees. Such a shortest path can be found by applying the Dijkstra [4] or A* [9] algorithms or with a heuristic approach, such as ant colony optimization (ACO) [5]. When edge costs can change, the naive approach is to recompute the shortest path from scratch. To avoid this, the D* algorithm [17] and its improved version D* Lite [11] are proposed which reuse previous results and, hence, reduce the necessary computations significantly. Instead of distances, general edge costs are often minimized. When regarding more than one optimization objective, these costs can be a weighted sum of multiple cost components. In [3,16,20], such composite costs are used to optimize path planning on a macroscopic evacuation graph model.
Path planning based on local observations and decisions is presented in [19]. Pedestrians are modeled to leave a building on the shortest path, unless they have to wait for a certain time in a jamming queue. Then, they start observing the traffic at alternative doors and change their route choice if it seems worth it. In [15], a congestion avoidance approach for traffic based on local observations is introduced. A network of traffic lights observes the traffic flow at intersections and exchanges this information. Cars are routed on a next-hop basis to the intersection with the best traffic-flow on the way to their destination. A similar approach is presented in [8], where sensor nodes observe risk factors, exchange

this information, and send evacuees to the next best sensor node on the way to the building's exit. A scenario in which mobile devices are used for observation of an evacuation scenario is considered in [13]. Mobile devices collect information about the locations of other mobile devices in the building. This information is used to construct a macroscopic evacuation graph model and the approach from [12] is used to solve the quickest flow problem. The mobile devices then select evacuation paths that correspond to their current location.

3 D* Lite

The D* Lite algorithm [11] is performed on a graph representation $G = (N, E)$ of the environment, which is different from the macroscopic evacuation graph described before. The environment is divided in a grid of patches (nodes N) and adjacent patches in the von-Neumann neighborhood, i.e. surrounding patches without the diagonal patches, are connected in the graph via edges $E : (N \times N)$. Edges have costs $c(e)$ assigned and D* Lite computes the path from a start patch $(n_s \in N)$ to a defined target patch $(n_t \in N)$ with minimal costs. The search starts at the target patch and is directed towards the start patch with the help of a heuristics value which is mostly the Euclidean distance between each patch $n \in N$ and the start patch $h(n)$. The procedure of D* Lite is described in the following.
The algorithm computes values $d(n)$ for $n \in N$ which represent the traveling costs from n to n_t until $d(n_s)$ is determined, starting at the target patch n_t. The shortest path can subsequently be derived by following the lowest cost edges. Let $U \subset N$ be a sorted set in the sense that all nodes $n \in U$ are sorted in ascending order with respect to $key_1(n) = min(d(n), \bar{d}(n)) + h(n)$ and, if two elements are equal, according to $key_2(n) = min(d(n), \bar{d}(n))$. Let $U.first$ denote the first element in this sorted set U, i.e. the element with minimal key value. We further assume that $Adj(n)$ returns all adjacent nodes of a node n with respect to the von-Neumann neighborhood. Algorithm 1 shows the procedure to compute the values $d(n)$ for all patches on the path with lowest traveling costs between n_s and n_t. The values $\bar{d}(n), n \in N$ are auxiliary variables.

If a change in costs is detected for a certain edge, the update procedure in Algorithm 1 is called for all affected nodes and for all nodes in list U recursively. The consistency check between $d(n)$ and $\bar{d}(n)$ in line 5 of procedure $update(n)$ ensures that the updates are constrained to the affected area and do not spread to all subsequent patches.

4 Evacuation Path Planning

In this section, we propose the DMO-DEP approach for computation of navigation instructions on mobile devices during a building evacuation. Each device is assumed to know the building's layout, e.g. by downloading it when entering the building, and computes an optimal evacuation route from it's current position

Algorithm 1: D* Lite

Input: graph $G = (N, E)$, start node n_s, target node n_t, heuristic $h(n) \forall N$, costs $c(e) \forall E$

```
d(n) = ∞ ∀n ∈ N, d̄(n) = ∞ ∀n ∈ N, d̄(n_t) = 0
U.add(n_t)
computePath()

Procedure computePath()
    while (key(U.first) < key(n_s)) ∨ (d̄(n_s) ≠ d(n_s)) do
        Define u = U.first
        if d(u) > d̄(u) then
            d(u) = d̄(u)
            update(u)
        end
        d(u) = ∞
        for p ∈ Adj(u) do
            update(p)
        end
    end

Procedure update(n)
    if n ≠ n_t then
        d̄(n) = min_{∀s∈Adj(n)}(c(n, s) + d(s))
        U.add(n)
        if d(n) = d̄(n) then
            U.remove(n)
        end
    end
```

to an exit of the building according to various objectives. Information about the current evacuation situation is gathered via local communication between nearby devices and with the help of the mobile ad hoc network that arises from these communication links. The information is used to assign costs to edges in the graph model of the environment described in Section 3 and the dynamic path planning algorithm D* Lite (cf. Section 2) is applied to find the minimum cost path for each mobile device. The computed evacuation path can be displayed on the device's screen as navigation instructions. We concentrate on path planning and assume that each device knows its localization, for example, with the help of an indoor localization system [7]. The costs that are minimized consist of various cost components, which are weighted with evacuee-specific weights, thereby, respecting characteristics of the evacuees in the relative importance of each cost component.

The patches $n \in N$ in the environment model have approximately the size of an average person and all patches in the von-Neumann neighborhood are connected via bidirectional edges $e \in E : (N \times N)$. A mobile device a can assign costs to each edge $e = (n, m)$, which describe the costs for the device's user to walk from patch n to m. These costs are described as a weighted sum of cost components for each considered optimization criterion $o \in O$:

$$c^a(e) = \sum_{o \in O} (w_o^a \cdot c_o^a(e)) \tag{1}$$

with weights $w_o^a \in [0, 1]$ and $\sum_O w_o^a = 1$. The weights can be different for each device depending on its user's characteristics. They can for example be derived from profile information about the device's user, such as age or handicaps, which are manually configured. However, we leave the challenge of finding optimal weights to future work. When the costs for all edges in the discretized layout are known, the D* Lite algorithm is used to find the minimum cost path, i.e. the sequence of patches $< n_s, ..., n_t >$. From the known sequence of patches, a sequence of rooms is extracted which we call evacuation instruction. A path between these rooms can then be displayed on the screen of the mobile device in order to guide the user to an exit. The transformation from patches to rooms might not be necessary, but it ensures that the displayed path is the shortest path between rooms instead of a possibly winding sequence of patches.
One obvious cost component that has to be regarded for evacuation path planning is the traveling distance for an edge $c_{distance}(e)$. This costs component is equal for all devices $a \in A$ and corresponds to the size of each patch. Apart from the traveling distance, we concentrate on two other objectives for choosing evacuation routes: risk minimization and congestion avoidance. For this, we define cost components that reflect the risk an evacuee is exposed to on a route towards the exit, as well as the congestion potential on this path.

4.1 Risk Minimization

In order to avoid risky paths, we assume that there are sensors inside the building which can measure potential danger indicators such as gas concentration or heat, similar to the scenario examined in [8]. The sensors forward a risk level $risk(R) \in [0, 1]$ for the room R which they observe to all devices that enter the room using the same means of communication that the mobile devices use among themselves. This information has an *age* value assigned to it that is initially set to zero when the message is received from a sensor node and increased according to the time that passes on the mobile device's system clock. All devices constantly disseminate detected risk information in the ad hoc network and whenever this information is forwarded to other mobile devices, the current *age* is also communicated. This ensures that all devices keep the most recent information in case they are informed about the risk level of a room via different sources. The risk level cost component $c^a_{risk}(e)$ is assigned to all edges $e(n, m)$ between nodes in the affected room R. By assigning the risk value to all patches in a room, it becomes comparatively more expensive to cross a larger room with the same risk level.

$$c^a_{risk}(e(n,m)) = risk(R), \forall n \in R \tag{2}$$

4.2 Congestion Avoidance

One objective of DMO-DEP is to use the information about other evacuees' locations in the building to evaluate evacuation paths with respect to their congestion potential. While the congestion prediction and avoidance is a well-known task in traffic optimization, it is a much more challenging task in a pedestrian scenario. This is mainly due to the comparatively unorganized situation that arises when evacuation is accompanied by panic, which makes it hard to predict the evacuation process [10]. This challenge is further aggravated by the uncertain information situation due to communication delays and link breakages in the network. As a consequence, predicting waiting times on a certain route is almost impossible. Fortunately, it is enough for evacuation planning to be able to compare two possible routes with each other. For this, we propose two indicators that are not meant to estimate the time evacuees need to travel on a certain route but to give information about potential congestion formations. For this, each mobile device constantly disseminates its own location information in the ad hoc network and collects these information from other devices. An *age* value is assigned to each information which is set to zero for the own location and incremented according to the devices' system clocks when forwarded in order to be able to identify more recent information.

Load The first congestion indicator is derived from the idea that many evacuees on a smaller area are more likely to jam up in front of exits. Therefore, we propose *load* costs which are computed for each edge $e(n,m)$ in a room $n, m \in R \subset N$ as the number of devices $d \in D$ that are located in this room with respect to the size of the room as follows:

$$c^a_{load}(e(n,m)) = \frac{|\{d \in D : n(d) \in R\}|}{|\{n \in R\}}|, \forall n \in R \tag{3}$$

with $n(d)$ referring to the patch on which the device d is located. One could argue that relating the number of devices in a room to the size of the room's doors is more effective, but a room can have multiple doors and yet only one determines the flow rate. Since it is hard to tell which doors will be used by how many evacuees, the described *load* parameter is the more straightforward congestion indicator.

Entropy Congestion is an emergent phenomenon, i.e. a formation of order from disorder based on self-organization, and as such is measurable by applying the concept of entropy, as suggested in [14]. Entropy is a well-known metric to measure the amount of order in a system. Low entropy is equivalent to a higher system order and vice versa. Hence, in a room where a congestion builds up, the locations of the corresponding mobile devices are distributed unevenly and the respective entropy value would be low. To compute the entropy of the evacuee' locations in a room the following considerations are made. Patches in a room can be organized in rows and columns reflecting their horizontal and vertical order respectively. Let a room R contain $x \times y$ patches, i.e. x columns and y rows. Let further $num(i)$, $num(j)$ denote the number of devices $d \in D$ located on column i or row j respectively. The entropy of a room R is computed as $e(R) = -\sum_{n \in R} p(n) \mathrm{ld}\ p(n)$ with $p(n) = \frac{num(x(n))+num(y(n))}{\sum_{i=1}^{x}\sum_{j=1}^{y} num(i)+num(j)}$ and $x(n)$, $y(n)$ denoting the row and column of a patch n.

To infer the corresponding costs for rooms with high agent concentration, the entropy is normalized to lie in the range $[0, 1]$ using the maximum entropy $e_{max}(R) = ld(x \cdot y)$ of a room R, when it is completely occupied by agents, and the minimum entropy value $e_{min}(R) = ld(x + y) - \frac{2}{x+y}$, when a single agent is in the room. The costs are then computed as:

$$c^a_{entropy}(e(n, m)) = 1 - (\frac{e(R) - e_{min}(R)}{e_{max}(R) - e_{min}(R)}), \forall n \in R \tag{4}$$

5 Experimental Study

In this section, the DMO-DEP approach is tested in a simulative experimental study. We want to verify, whether the congestion indicators are useful to avoid jammed evacuation routes and whether the risk avoidance mechanism works.

5.1 Evacuation Simulation

For the experimental study we use a simple evacuation model similar to the two-dimensional cellular automaton described in [2]. The simulation environment is shown in Figure 1. In each time step, an agent exchanges messages with nearby devices to collect information, computes its optimal evacuation path, and is allowed to move to an accessible adjacent patch in its von-Neumann neighborhood. The evacuee follows the shortest path measured in Manhatten distance metric that connects its current patch with the closest patch belonging to the next room according to the evacuation path computed via DMO-DEP. If there is no free patch in the von-Neumann neighborhood but in the Moore-neighborhood, i.e. all adjacent patches including the diagonal patches, and this patch is closer to the target than the current patch measured in Euclidean distance, the individual moves in two time steps to this patch in the Moore neighborhood. This has the effect that agents cluster in front of doors in an arc-like shape instead of lining-up in front of it. In each simulation cycle all agents are executed once in random order.

5.2 Evaluation of Risk Minimization

To evaluate the DMO-DEP approach, we start with a scenario to test the risk avoidance of the evacuees. For this, the risk level in room 4 and 10 of the building

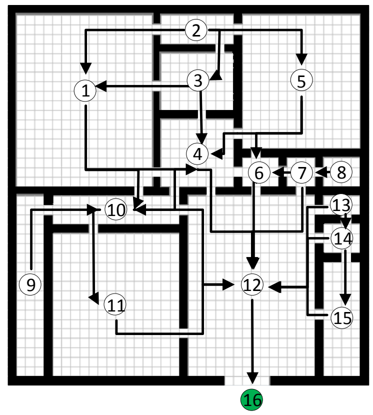


Figure 1: Building layout used for the experiments.

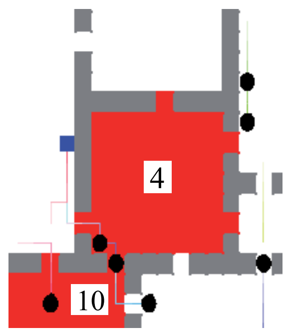


Figure 2: Risk avoidance for two different agent classes. More risk aversive agents are depicted in black.

layout are set to a value of 1 and two different classes of evacuees are simulated. One which is more risk aversive with weights $w_{risk} = 0.9$, $w_{distance} = 0.1$ and less careful agents with $w_{risk} = 0.8$ and $w_{distance} = 0.2$. Figure 2 shows a sample situation during an experiment run, in which the mechanisms for risk avoidance can be observed. The agent depicted as a blue square is a risk aversive agent. The snapshot shows that the risk-aversive evacuee makes a detour to avoid room 4, while the agents with higher risk tolerance pass through the rooms with higher risk level. Also, the dissemination of risk information can be observed in this simple example because the risk-aversive agent did not need to go inside room 4 to know about its risk level, but was informed about this by other agents.

To get an insight about the impact of risk avoidance on the evacuation time, we repeat this experiment with 60 randomly located agents and set the weights in the cost function to $w_{risk} = 0.9$, $w_{distance} = 0.1$ for all agents. Table 1 displays the average results for 40 repetitions. This evaluation reveals an intuitively expected effect. The reduced average time that an agent spends

Table 1: Numerical evaluation of risk avoidance.

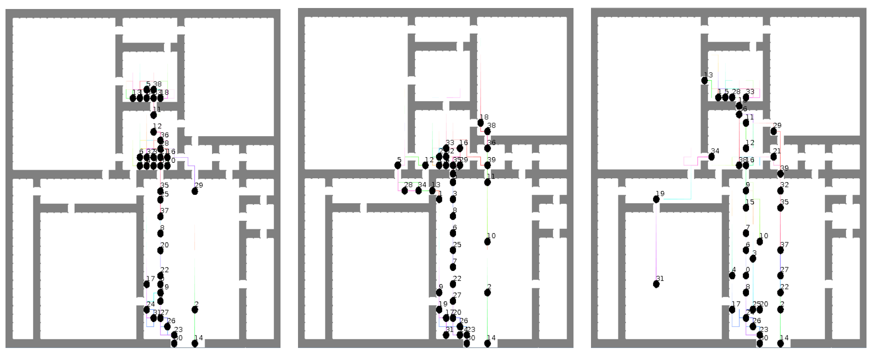
DMO-DEP:	distance	risk
total evac. time	59.1	89.6
avg. evac. time	31.0	36.2
avg. risk time	2.8	0.8
avg. waiting time	0.5	0.7

in rooms with higher risk levels comes at the cost of an increased evacuation time. Simultaneously, the average waiting time per agent is higher because more evacuees take less risky routes towards the exit which increases the congestion potential on these paths.

5.3 Evaluation of Congestion Avoidance

The next experiment tests the impact of the two proposed congestion indicators on the evacuation performance. For this, we first consider an evacuation scenario in which agents are distributed across the two rooms with number 3 and 4. Congestions will occur in both rooms when the agents only consider the distance as an optimization criterion. Figure 3 illustrates a sample situation that occurs when only distance costs are optimized (a), with additional load costs (b), and with entropy costs (c). The weights for scenarios (b) and (c) are chosen as $w_{load} = w_{entropy} = 0.8$ and $w_{distance} = 0.2$. From these snapshots, it can be observed that both congestion indicators prevent the emergence of jamming queues. Additionally, it becomes apparent that entropy and load costs influence the behavior of the evacuees in different ways. While minimizing load costs keeps up a high concentration of agents in the building, this is not the case when entropy costs are considered. In Figure 4, the numerical evaluation of the three depicted experiment runs is shown which mainly confirms the previous observations. Both, the integration of entropy and load costs reduce the overall evacuation time by 23% for the experiment with entropy costs and 33% for the experiment with load costs. Also, the average evacuation time per evacuee and the average waiting time, i.e. cycles without movements per agent, are reduced significantly.

To confirm the generality of these results and to compare them with the performance of the capacity constrained distributed swarm evacuation approach (CCRP) presented in [13], we test an evacuation scenario with 60 evacuees, which are randomly distributed in the building and perform 40 repetitions. Costs are set to $w_{load} = w_{entropy} = w_{distance} = 0.5$. The macroscopic evacuation graph for CCRP is depicted in Figure 1. Table 2 shows the results. DMO-DEP improves the total evacuation time compared to the CCRP approach by one round on average and the average evacuation time by three rounds, even without congestion indicators. The reason for this lies in the less abstract evacuation model where distances are reflected more precisely and edges can are bidirectional compared to the macroscopic graph representation used for CCRP. The



(a) Distance costs (b) Load costs (c) Entropy costs

Figure 3: Evacuation situation after 24 cycles with only distance costs (a), distance and load costs (b) and distance and entropy costs (c) to optimize.

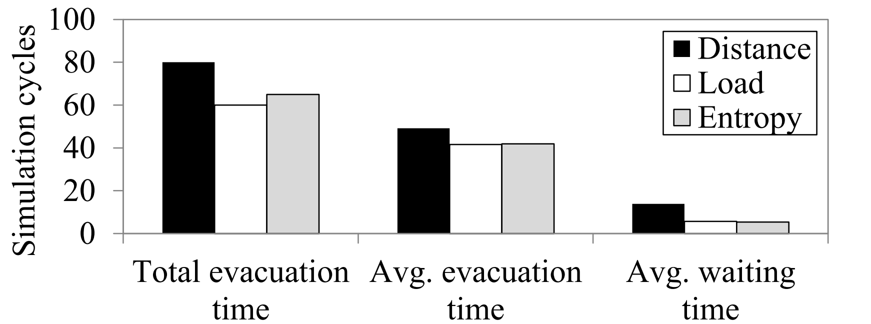


Figure 4: Numerical evaluation of the simulations depicted in Figure 3

Table 2: Comparison of DMO-DEP with and without congestion indicators and CCRP.

	distance	load	entropy	CCRP
total evac. time	58.1	59.4	59.1	60.3
avg. evac. time	30.8	31.2	31.0	32.6
avg. waiting time	0.5	0.3	0.5	2.5

average waiting time shows that CCRP navigation instructions expect more evacuees to wait at jammed doors compared to DMO-DEP, which makes DMO-DEP more attractive for evacuation planning since waiting in panic situations is unnatural. The reason why the improvements of DMO-DSEP are relatively subtle, lies in the random initial distribution of agents which does not lead to massive congestion emergence compared to the scenario in Figure 4. Integration of the load indicator further reduces the total evacuation time by one round on average, which confirms its effectiveness. In contrast to load, entropy-based congestion indication increases the evacuation time with simultaneous decrease in waiting time. The reason for this lies in the avoidance of high agent concentrations. Evacuation can sometimes be faster when evacuees wait in small jamming queues in front of doors instead of always preferring detours. We conclude from this that entropy-based congestion indication is effective to avoid jamming queues, but has to be applied with caution.

6 Conclusion and Future Work

We present DMO-DEP, an approach to compute optimal evacuation paths with respect to several objectives in order to use mobile devices for evacuation support. The capability to form ad hoc network connections and exchange information is used to avoid congestions during the evacuation process. The application of a dynamic path finding algorithm allows for fast replanning of navigation instructions, incorporating newly available information in the path planning process and, hence, make it adaptive. It is shown that even though the information from the ad hoc network communication is uncertain, it can be used to evaluate different routing options and it is sufficient to speed up the evacuation process compared to a situation without information exchange. With DMO-DEP, a decrease in evacuation time compared to the CCRP approach from [13] is achieved. In addition, the average waiting time per agent can be reduced significantly, which makes it more likely for evacuees to follow the navigation instructions of DMO-DEP. We investigated two congestion indicators based on the number and concentration of evacuees in rooms, as well as a risk aversion strategy. It is shown that DMO-DEP is able to integrate agent specific characteristics in the navigation planning, as well as multiple objectives. Both indicators were shown to be successful in the prevention of jamming queues, but the entropy based indicator slows down the evacuation process due to the high preference of detours, and has to be applied with caution. For future work, we want to concentrate on finding a way to learn the best weights for congestion indication for specific buildings. Additionally, it is important to investigate how uncertainty in location information influences the results.

References

[1] R. Burkard, K. Dlaska, and B. Klinz. The quickest flow problem. *Mathematical Methods of Operations Research*, 37(1):31–58, 1993.

[2] C. Burstedde, K. Klauck, A. Schadschneider, and J. Zittartz. Simulation of pedestrian dynamics using a two-dimensional cellular automaton. *Physica A: Statistical Mechanics and its Applications*, 295(34):507–525, 2001.

[3] N. Cheng. An optimization method for dynamic evacuation route programming based on improved ant colony algorithm. In *International Conference on Intelligent System Design and Engineering Application*, volume 1, pages 265–267. IEEE Computer Society, 2010.

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

[5] M. Dorigo, V. Maniezzo, and A. Colorni. Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 26(1):29–41, 1996.

[6] R. F. Fahy and G. Proulx. Analysis of published accounts of the world trade center evacuation. federal building and fire safety investigation of the world trade center disaster, 2010.

[7] N. Fallah, I. Apostolopoulos, K. Bekris, and E. Folmer. Indoor human navigation systems: A survey. *Interacting with Computers*, 25(1):21–33, 2013.

[8] A. Filippoupolitis, G. Gorbil, and E. Gelenbe. Spatial computers for emergency management. In *IEEE Conference on Self-Adaptive and Self-Organizing Systems Workshops*, pages 61–66, 2011.

[9] P. E. Hart, N. J. Nilsson, and B. Raphael. Correction to "A formal basis for the heuristic determination of minimum cost paths". *SIGART Bull.*, (37):28–29, 1972.

[10] D. Helbing, I. Farkas, and T. Vicsek. Simulating dynamical features of escape panic. *Nature*, 407(6803):487–490, 2000.

[11] S. Koenig and M. Likhachev. Improved fast replanning for robot navigation in unknown terrain. In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, volume 1, pages 968–975, 2002.

[12] Q. Lu, B. George, and S. Shekhar. Capacity constrained routing algorithms for evacuation planning: A summary of results. In *Proceedings of the 9th International Symposium on Spatial and Temporal Databases*, pages 22–24, 2005.

[13] S. Merkel, S. Mostaghim, D. Blum, and H. Schmeck. Distributed swarm evacuation planning. In *Proceedings of 2013 IEEE Symposium Series on Computational Intelligence*. IEEE Computer Society, 2013. (accepted).

[14] M. Mnif and C. Mller-Schloer. Quantitative emergence. In C. Mller-Schloer, H. Schmeck, and T. Ungerer, editors, *Organic Computing A Paradigm Shift for Complex Systems*, number 1 in Autonomic Systems, pages 39–52. Springer Basel, 2011.

[15] H. Prothmann, S. Tomforde, H. Lyda, H. Branke, H. Hhner, C. Mller-Schloer, and H. Schmeck. Self-organised routing for road networks. In F. A. Kuipers and P. E. Heegaard, editors, *Self-Organizing Systems*, number 7166 in LNCS, pages 48–59. Springer Berlin Heidelberg, 2012.

[16] M. Saadatseresht, A. Mansourian, and M. Taleai. Evacuation planning using multiobjective evolutionary optimization approach. *European Journal of Operational Research*, 198(1):305–314, 2009.

[17] A. Stentz and I. Mellon. Optimal and efficient path planning for unknown and dynamic environments. *International Journal of Robotics and Automation*, 10:89–100, 1993.

[18] J. Szwedko, C. Shaw, A. G. Connor, A. Labrinidis, and P. K. Chrysan. Demonstrating an evacuation algorithm with mobile devices using an e-scavenger hunt game. In *Proceedings of the 8th ACM International Workshop on Data Engineering for Wireless and Mobile Access*, pages 49–52, 2009.

[19] A. U. K. Wagoum, A. Seyfried, and S. Holl. Modelling dynamic route choice of pedestrians to assess the criticality of building evacuation. arXiv e-print 1103.4080, 2011.

[20] X. Zong, S. Xiong, Z. Fang, and Q. Li. Multi-objective optimization for massive pedestrian evacuation using ant colony algorithm. In Y. Tan, Y. Shi, and K. C. Tan, editors, *Advances in Swarm Intelligence*, volume 6145, pages 636–642. Springer Verlag, 2010.