EVACUATION DYNAMICS INFLUENCED BY SPREADING HAZARDOUS MATERIAL

SIMONE GÖTTLICH
School of Business Informatics and Mathematics
University of Mannheim
D-68131 Mannheim, Germany

SEBASTIAN KÜHN, JAN PETER OHST, STEFAN RUZIKA
AND MARKUS THIEMANN
Department of Mathematics
TU Kaiserslautern
D-67663 Kaiserslautern, Germany

Abstract. In this article, an evacuation model describing the egress in case of danger is considered. The underlying evacuation model is based on continuous network flows, while the spread of some gaseous hazardous material relies on an advection-diffusion equation. The contribution of this work is twofold. First, we introduce a continuous model coupled to the propagation of hazardous material where special cost functions allow for incorporating the predicted spread into an optimal planning of the egress. Optimality can thereby be understood with respect to two different measures: fastest egress and safest evacuation. Since this modeling approach leads to a pde/ode-restricted optimization problem, the continuous model is transferred into a discrete network flow model under some linearity assumptions. Second, it is demonstrated that this reformulation results in an efficient algorithm always leading to the global optimum. A computational case study shows benefits and drawbacks of the models for different evacuation scenarios.

1. Introduction. An evacuation process in an emergency situation can be considered as the task of moving residents quickly out of a danger zone to safety. Evacuation modeling is an important tool for estimating the total evacuation time, identifying dangerous substructures in the considered area, and optimizing evacuation strategies (see [34, 35, 18, 14, 38]). Evacuation models are usually classified into microscopic and macroscopic approaches. Microscopic models simulate the behavior of each individual evacuee. Interaction between evacuees and local phenomena such as lane formation of pedestrians and bottlenecks can be identified in these models [35, 25]. Macroscopic approaches model evacuees as part of a homogeneous group and do not take into account individual behavior. Different aspects of an evacuation scenario such as infrastructure, evacuees, and safe locations are considered. With this overall knowledge, system optima yielding among others minimal evacuation times or fastest evacuation routes can be computed. Macroscopic models based on optimization techniques can be designed such that their results correspond
to a best-possible evacuation, that means, e.g., that the evacuation times computed are provable lower bounds on the true evacuation time [18, 15].

In most applications, macroscopic evacuation models are based on linear dynamic network flows. The big advantage of this modeling approach is the availability of fast and well-investigated algorithms. Dynamic network flow models are discrete macroscopic models that are capable of yielding optimal evacuation strategies [5, 4]. In dynamic networks, flow units are assumed to take time to traverse an arc [37]. This generalization of static network flows was first considered by Ford and Fulkerson [9] who also proposed a method to compute a maximal dynamic flow between two specified nodes using temporally repeated flows computed in the underlying static network.

For evacuation modeling, the quickest flow problem, where the goal is to find the minimum time to bring all flow units in the network to a predefined sink node, is also of importance. It can be solved in polynomial time by combining maximum dynamic flows with an appropriate binary search over the time horizon [3]. Extensions and variants of dynamic network flow problems have been examined to include phenomena emerging in evacuation processes. Hamacher and Tjandra [17] solved the earliest arrival problem with time-dependent capacities, enabling models with temporally failure of arcs. Flow and load dependent travel times reflect slow-down of evacuees on congested arcs and are considered in [27, 28]. Lately, microscopic aspects are included in macroscopic models as well as both, microscopic and macroscopic models, are combined in common framework to obtain more reliable information about the evacuation process [16, 26].

From a pde point of view, evacuation aspects can be found in the analysis of pedestrian movements, see [19, 22, 23] for an overview. The majority of these models is formulated on 1d-or 2d corridors but not on networks. To map pdes on networks, it is necessary to define coupling conditions at nodes ensuring the conservation of mass. For instance, this is done in traffic flow [6, 20] or supply chain modeling [12, 7]. Both approaches are close to the pedestrian models and therefore many ideas can be copied.

In this work, we are interested how the propagation of some hazardous material (e.g. a gas cloud) influences the evacuation process. This means, the evacuation model under consideration is combined with an equation for the hazardous material in a clever way. To solve the coupled problem, we derive a new framework and apply the aforementioned dynamic network flow algorithms. Further, we investigate the numerical properties of the network flow approach by comparing computing times for the solution to the standard mixed–integer approach and demonstrate the efficiency of network flow algorithms in this context.

The remainder of this article is organized as follows. In Section 2, we start from a pde-based evacuation model which is coupled to the expansion of spreading hazardous material described by an advection-diffusion equation. In Section 3 we introduce two solution procedures for the coupled problem: on the one hand a straightforward discretization leading to a mixed–integer model and, on the other hand, a reformulation as a dynamic network flow problem. The numerical results are presented in Section 4 and contain comparisons of the two solution approaches and case studies as well.

2. Continuous evacuation model. To introduce an evacuation model that is suitable for regional evacuation purposes, we need a mathematical definition of a
network where roads correspond to arcs and cities to nodes. In the beginning of the evacuation process, all evacuees are located in a source node from where they shall get to safety. Therefore, we define the following:

Definition 2.1 (Network definition). For a time horizon $T$ let $G = (V, A)$ be a directed graph with node set $V$ and arc set $A$. The node set is assumed to subsume a source $s \in V$ and a sink $d \in V$. For a node $i \in V$, $\delta^-(i)$ and $\delta^+(i)$ is the set of all arcs starting and ending at $i$, respectively. An arc $e \in A$ is associated with an interval $[a_e, b_e]$ of length $L_e$. The constant times needed to traverse one arc are called travel times $\tau_e$ and correspondingly the travel velocities are $v_e = \frac{L_e}{\tau_e}$. Arc capacities are denoted by piecewise constant functions $\mu_e(t, x)$ and storage capacities at nodes by time-dependent functions $M_e(t)$.

Remark 1. Mathematically, the model is restricted to subsume one source and one sink only. Note that the model can easily be extended to multiple sources and multiple sinks using a super source and a super sink, respectively.

For our modeling, we prefer a macroscopic model composed of ordinary and partial differential equations. Similar to the ideas used in traffic flow network models [6, 21] and the supply chain approach [12, 7], we stick to the following pde/ode-restricted optimization problem describing the evacuation scenario:

\[
\begin{align*}
\min / \max & \Phi(\rho, q) \\
\text{s.t.} & \partial_t \rho_e(t, x) + \partial_x f_e(\rho_e(t, x)) = 0 \quad \forall e \in A, x \in [a_e, b_e], t \geq 0 \quad (1b) \\
& \text{where } f_e(\rho_e(t, x)) = v_e \rho_e(t, x) \\
& \partial_t q_i(t) = \sum_{e \in \delta^-(i)} f_e(\rho_e(t, b_e)) - \sum_{\tilde{e} \in \delta^+(i)} f_{\tilde{e}}(\rho_{\tilde{e}}(t, a_{\tilde{e}})) \quad \forall i \in V, t \geq 0 \quad (1c)
\end{align*}
\]

\[
\begin{align*}
q_i(0) &= q_{i,0} & \forall i \in V \quad (1d) \\
0 \leq q_i(t) & \leq M_e(t) & \forall i \in V, t \geq 0 \quad (1e) \\
0 \leq f_e(\rho_e(t, x)) & \leq \mu_e(t, x) & \forall e \in A, x \in [a_e, b_e], t \geq 0. \quad (1f)
\end{align*}
\]

The objective function (1a) must be chosen in an appropriate manner. Note that the function $\Phi$ might depend on the density of evacuees $\rho_e$ on arcs $e \in A$ and the load of the queue $q_i$ in vertices $i \in V$, where the queues buffer an excess of net inflow, cf. [2]. Possible choices of the objective function in the context of evacuation are discussed below and in Subsection 2.1. The advection equation (1b) describes the movement of people on arcs and ensures conservation of mass. We additionally assume initial values $\rho_e(0, x) = \rho(x)_{e,0}$. Usually, depending on the application, it is necessary to prescribe boundary data in form of an influx for $e \in \delta^+(s)$. In this model, the relation between the flux $f_e$ and the density $\rho_e$ is simply linear, i.e. $f_e = v_e \rho_e$. That means, natural bottleneck situations as in traffic flow models cannot occur. But as we will see later on, this linear structure will help to interpret the evacuation model in the sense of dynamic network flow models. Equation (1c) determines the load of the queue $q_i$ in node $i$. The ordinary differential equation is a rate equation measuring the total ingoing $\sum_{e \in \delta^-(i)} f_e$ and outgoing flux $\sum_{\tilde{e} \in \delta^+(i)} f_{\tilde{e}}$ into this node. Here, we have degrees of freedom to distribute the flux. It is the subject of optimization to determine the best routing of evacuees through the network. The initial load of each queue $i \in V$ is given by $q_{i,0}$ representing the number of residents in vertex $i$ to be evacuated, see equation (1d).
Box constraints (1e) and (1f) are bounded by the respective capacity constraints from above.

Remark 2. The problem (1) is inspired by the evacuation model motivated in [18]. Due to its linear nature, it is the simplest approach to model an evacuation process. This means, either the evacuees move forward towards their safety destination as long as capacity constraints permit escape or, if the capacity is reached, the evacuees have to stay in nodes. Therefore, congestions on arcs are avoided. However, a more sophisticated model should allow for bottleneck situations on arcs, cf. traffic flow models [6, 21].

From literature about evacuation modeling, we know that for a fixed time horizon $T$, the goal of the so-called maximum dynamic flow problem is to send the maximum possible number of flow units from source to sink. The side constraints to this problem are conservation of mass (1b), (1c) and capacity restrictions (1e), (1f) with $q_{i,0} = 0$ for all $i \in V$. The goal is to optimize the following objective function

$$\max \Phi(\rho, q) = \int_0^T \sum_{e \in \delta^-} f_e(\rho_e(t, a_e)) \, dt.$$  

(2)

However, for a given amount of flow, the quickest flow problem seeks for the minimum time horizon to send all flow units from source to sink through the network. This can be modeled using (1b) – (1f) with $q_{i,0} = 0$ for all $i \in V \setminus \{s\}$ and $q_{s,0} > 0$ and the following objective function

$$\min \Phi(\rho, q) = \sup \left\{ t \in [0, \infty) \mid f_e(\rho_e(t, b_e)) \neq 0 \text{ for some } e \in \delta^-(d) \right\}. \tag{3}$$

Further, it has to be ensured that all evacuees are indeed sent through the network. Thus, an additional condition is introduced

$$\int_0^T \sum_{e \in \delta^-} f_e(\rho_e(t, b_e)) \, dt = q_{s,0}. \tag{4}$$

The main focus of this work is to couple the evacuation model (1) to the spread of some hazardous material (think of a gas cloud). We present a two-stage modeling approach where we first compute the propagation of gas by an advection-diffusion equation in 2d and second map the concentration as extra costs onto the network. The resulting evacuation model is supplemented with a new objective function involving concentration-depending costs.

2.1. Evacuation influenced by hazardous material. Since the task of evacuating people from a certain area has been intensely studied in the past (see e.g. [18, 13]), we consider an advanced scenario where time-dependent parameters enter the problem due to an external source, which will be called hazardous material. This may be a cloud consisting of gas, radioactive material (e.g. when considering the evacuation of the area around a nuclear power plant after an MCA), or an ash cloud from a volcanic eruption. In doing so, we first introduce the equations and parameters describing the hazardous material. Then we discuss possibilities to map the new information onto the network. Finally, we introduce the coupled evacuation problem.

The hazardous equation shall set up additional parameters for the model (1). The original model relies on a general graph. Yet, in view of the application, vertices are
specified by coordinates in $\Omega \subset \mathbb{R}^2$. Furthermore, we assume that the concentration of hazardous material $\omega$ obeys the following linear advection-diffusion equation:

$$\partial_t \omega - \kappa \Delta \omega + d \cdot \nabla \omega = 0 \quad \text{in } \mathbb{R}^+ \times \Omega$$  \hspace{1cm} (5)

with diffusion constant $\kappa > 0$ and wind vector field $d = (d_1, d_2) \in \mathbb{R}^2$. The starting level of the concentration is 100\% (life-threatening value) in the area of the source and 0\% elsewhere. On the boundary of $\Omega$, we use Dirichlet conditions with $\omega = 0$. Note that this way of modeling does not include a time-dependent source and thus, the concentration is fixed to the amount given at the beginning of the calculation.

**Remark 3.** Since the focus of this subsection is on the mapping of equation (5) on the network, we prevent a discussion on more realistic propagation models.

Due to the simple dynamics of the hazardous material (associated with standard numerics), we now turn our attention to the mapping of the values $\omega$ onto the network. We present two ways to include the concentration level as an additional parameter in the evacuation model (1) which lead to a coupling to the original model via a new objective function $\Phi$.

**Remark 4.** An alternative coupling might be to adjust the capacities $\mu_e(t, x)$, $M_e(t)$ or travel velocities $v_e$ of arcs. E.g. in exposed areas the number of accidents may increase since people are getting nervous. This potentially leads to smaller capacity constraints and lower velocities.

In the following, we concentrate on a new objective function approach and introduce time-dependent concentration parameters $c_e(t)$ and $c_i(t)$ defined on arcs $e$ and vertices $i$, respectively. We map the concentration value $\omega$ to the corresponding positions in the network, e.g. $\omega|_e$ means that the values of $\omega$ are restricted to the arc $e$, and comment on two integral functionals, called static model and dynamic model.

1. **Static model:** The concentration in the network is solely considered at a certain point in time $t$. The value of the hazardous material on an arc $e$ at time $t$ is normalized and multiplied by the travel time $\tau_e$ yielding

$$c_e(t) = \tau_e \int_a^b \omega|_e \frac{L_e}{dx} \, dx.$$  \hspace{1cm} (6)

The travel time is included in the computation of the cost, because this displays that evacuees on an arc with large travel time are more exposed to the concentration than evacuees on arcs with smaller travel times.

For a vertex $i \in V$ the concentration parameter at time $t$ represents the exposure of evacuees to the hazardous material when waiting in the vertex for one time step. Thus, we get

$$c_i(t) = \omega|_i.$$  \hspace{1cm} (7)

2. **Dynamic model:** This model includes, in contrast to the static model, the whole dynamics of the hazardous material in the calculation of the parameter $c_e(t)$. The static model does not take into account that the level of the hazardous material may change during the time the evacuees traverse an arc. This property is included by the following choice of the concentration parameter, regarding all times within the interval $[t, t + \tau_e]$

$$c_e(t) = \int_t^{t+\tau_e} \int_{a_e}^{b_e} \frac{\omega|_e}{L_e} \, dx \, dt.$$  \hspace{1cm} (8)
Correspondingly, for a vertex $i$, we get
\[
c_i(t) = \int_t^{t+1} \omega_{i} d\tilde{t}.
\] (9)

Naturally, the dynamic model is closer to reality than the static model. But obviously its drawback is the need of the full information on the hazardous material in advance. Thus, the dynamic model does not allow for an online use. Numerical experiments comparing the static and the dynamic model can be found in Section 4.

An appropriate possibility for including the costs in the evacuation model is to minimize the level of concentration to which the evacuees are exposed. This can be done with the definition of new cost-dependent objective functions $\Phi(\rho, q, c)$. On the one hand the maximal level of concentration to which the evacuees are exposed is minimized by
\[
\Phi(\rho, q, c) = \max \left\{ \max_{\forall t \in [0,T], \forall x \in [a_e,b_e], \forall e \in A | \rho_e(t,x) > 0} c_e(t), \max_{\forall t \in [0,T], \forall i \in V | q_i(t) > 0} c_i(t) \right\}.
\] (10)

This objective function is sensible, if there is a level of concentration which directly causes the loss of evacuees, because the optimization seeks a solution avoiding arcs with high values of concentration costs. For example, if a hazardous gas is considered, which causes the death of evacuees if its concentration exceeds a certain percentage in the air but does no real harm below this critical percentage, this objective function will lead to an evacuation plan avoiding the highly contaminated arcs but therefore using – possibly more – arcs with concentration parameters slightly below the critical value.

On the other hand the total amount of concentration is minimized by
\[
\Phi(\rho, q, c) = \int_0^T \left( \sum_{e \in A} c_e(t) f_e(\rho_e(t,x)) + \sum_{i \in V} c_i(t) \partial_i q_i(t) \right) dt.
\] (11)

This objective function is useful when the hazard is harmful if the accumulated level exceeds a critical level (e.g. low levels of concentration). In summary, both objective functions can be used to extend the evacuation model (1) by solving the optimization problem
\[
\min (10) \text{ or } (11) \quad \text{s.t.} \quad (1b) - (1f). \tag{12a}
\]

Generally, this type of optimization problem can be solved using adjoint-based methods due to the continuous nature of the problem. In case of a linear objectives and constraints or only ‘mild’ nonlinearities, it is possible to reformulate the original problem into a mixed-integer program. The latter will be explained in Section 3.

**Remark 5.** In many applications, it is useful to consider the combination of both objective functions (10) and (11). This will lead to a so-called multi-objective optimization problem that requires more involved solution methods.

3. **Solution methods.** In Section 2 an evacuation model using partial and ordinary differential equations as constraints was introduced. The usual way to solve such an optimization problem is to calculate the adjoint equations as proposed in [24]. This is a straightforward computational approach first deriving the first order optimality system and then applying a gradient scheme to compute a (local) optimum. Since the evacuation problem is purely linear and the computation of
the optimality system is rather complex we prefer another approach following the ideas in [11, 10]. Therein, a mixed-integer program is introduced by discretizing the underlying network model using suitable numerical methods. This approach works for the traffic flow and supply chain model pretty well and can also be applied for the evacuation model (1). As it turns out, the evacuation model has nice features (linear dynamics on arcs, conservation of mass through nodes and box constraints) and, thus, it is even possible to put the model (1) into the framework of a dynamic network flow model, cf. [18]. This is a favorable approach, since dynamic network flows are well-investigated (see e.g. [1, 8, 18, 37]) and make use of very efficient algorithms leading to the global optimum.

3.1. Mixed-integer program. To derive the mixed-integer model, similarly to [10], we have to introduce new variables for the flux at the boundary of \([a_e, b_e]\) at a fixed arc \(e\) and a new variable for the queue for all time steps \(k\) of a time grid \(k = 0, \ldots, T\):

\[
\alpha^k_e := f_e(\rho_e(t, a_e)), \quad \beta^k_e := f_e(\rho_e(t, b_e)), \quad \gamma^k_e := q_e(t), \quad \forall e, k. \tag{13}
\]

As abbreviation for the original objective function \(\Phi(\rho, q, c)\), we use a straightforward discretized version \(\Phi_{\text{disc}}(\alpha, \beta, \gamma, c)\) depending on the respective variables. Then, the mixed-integer program (MIP) of the evacuation model (1) reads:

\[
\min / \max \Phi_{\text{disc}}(\alpha, \beta, \gamma, c) \tag{14a}
\]

\[
\text{s.t. } \beta^k_e = \alpha^{k-\tau_e} \quad \forall e \in A, \forall \tau_e \leq k \leq T \tag{14b}
\]

\[
\sum_{e \in \delta^+(i)} \beta^k_e = \sum_{e \in \delta^-(i)} \beta^k_e \quad \forall i \in V \setminus \{s, d\}, \forall 0 \leq k \leq T \tag{14c}
\]

\[
\gamma^{k+1}_e = \gamma^k_e + \Delta t (\alpha^k_e - \beta^k_e) \quad \forall e \in A \setminus \{(s, i) | i \in V\}, \forall 0 \leq k \leq T-1 \tag{14d}
\]

\[
\gamma^0_e = 0 \quad \forall e \in A \tag{14e}
\]

\[
0 \leq \alpha^k_e, \beta^k_e \leq \mu^k_e \quad \forall e \in A, \forall 0 \leq k \leq T \tag{14f}
\]

\[
0 \leq \gamma^k_e \quad \forall 0 \leq k \leq T, \forall e \in A \tag{14g}
\]

\[
0 \leq \gamma^k_e \leq M^k_e \quad \forall e \in A, \forall 0 \leq k \leq T. \tag{14h}
\]

Let us explain the equations in (14) step by step. A special choice of the objective function - corresponding to the maximum dynamic flow problem - is given by the maximization of flux reaching the sink

\[
\max \Phi_{\text{disc}}(\alpha, \beta, \gamma, c) = \sum_{k=0}^{T} \sum_{e \in \delta^-(d)} \Delta t \beta^k_e. \tag{14i}
\]

Consequently, the problem (14) reduces to a linear program (LP) since the objective functions and all constraints are linear. For the problem (14), we only focus to the max-flow objective, since we are interested in computing times of the LP compared to other approaches, see Section 4. Note that only depending on the objective function the problem (14) will become a mixed-integer program. E.g. the nonlinear objective function (10) requires binary variables making the problem a mixed-integer one. The transport equation (1b) is reformulated with the help of the Origin Tracking Algorithm [29, 30]. This algorithm has been invented to track down
the origin of contamination in water supply networks, but it is general in nature and we use it to detect the inflow on each arc. Using a unit time discretization and integer travel times simplifies the algorithm and allows for an exact calculation of the time step of the inflow. Then, equations (14b) and (14c) represent the exact solution which is independent of the CFL-condition. For the right implementation of the coupling condition (1c), we need variables $\zeta_k^e$ collecting all influx at $x = a_e$, see (14d). The original ordinary differential equation for the queue is discretized using the explicit Euler method (14e). The remaining constraints remain unchanged and can be discretized directly. Basically, the MIP (14) is solved using the commercial solver of CPLEX [39] which is based on Branch-and-Bound and Branch-and-Cut techniques. For a numerical case study, we refer to Section 4.

3.2. Dynamic network flow model. We start with a motivation for a dynamic network flow model. Typically, such a model is based on linear transport and conservation of mass. Capacity constraints are used to restrict the flow on arcs and nodes. Due the linearity of $f_e(\rho_e)$, the solution of the transport equation (1b) in (1) is known and we can write

$$\rho_e(t, x) = \rho_e(t - \frac{x - a_e}{v_e}, a_e) \quad \forall x \in [a_e, b_e], \quad t \geq \frac{x - a_e}{v_e}. \quad (15)$$

Furthermore, the flow conservation constraint (1c) in the evacuation model (1) can be rewritten as follows. The left-hand side $\partial_t q_i$ is approximated via the differential quotient whereas the right-hand side is reformulated applying (15) in combination with $\tau_e = (b_e - a_e)/v_e$, cf. Definition 2.1. Then, we end up with following equation for a fixed vertex $i$:

$$q_i(t + \Delta t) - q_i(t) = \sum_{e \in \delta^-(i)} \Delta t v_e \rho_e(t - \tau_e, a_e) - \sum_{e \in \delta^+(i)} \Delta t v_e \rho_e(t, a_e) \quad (16)$$

Now, we introduce the notation to translate equations (15), (16) in the sense of a typical dynamic network flow model. Let us assume that we have an integer time horizon $T \in \mathbb{N}$, an equidistant time-grid where $\Delta t = 1$ and we denote the time steps by $t_k$ with $0 = t_0$ and $t_{k_{\text{max}}} = T$. The variables $x_{ij}(t_k)$ describe the flow on arc $e = (i, j) \in A$ at time step $t_k$ and $y_i(t_k)$ the flow waiting in vertex $i \in V$ at time step $t_k$. As before, $\tau_{ij}$ describes the travel time on arc $e = (i, j)$ and correspondingly $a_i(t_k)$ and $b_i(t_k)$ are the discrete capacity limits. In contrast to the continuous evacuation model (1), the parameters $\tau_{ij}, a_i(t_k), b_i(t_k)$ are restricted to $\mathbb{N}$. To shorten notation we use $X = (x_{ij}(t_k))_{(i,j) \in A, 0 \leq k \leq k_{\text{max}}}$, $Y = (y_i(t_k))_{i \in V, 0 \leq k \leq k_{\text{max}}}$ for the whole flow. Furthermore we assume $x_{ij}(t_k) = 0$ for $t_k < 0$. The number of initial residents in each vertex $i$ is given as $y_{i,0} \geq 0$. The source $s \in V$ is the vertex in the danger zone, where all the evacuees are located at the beginning of the evacuation process. Thus, as we consider only a single source problem, $y_{s,0} > 0$
is the only non-zero value in \((y_{i,0})_{i \in V}\). Summarizing, we state:

\[
\begin{align}
\min / \max \phi(X, Y) \\
\text{s.t. } & y_i(t_k) - y_i(t_{k-1}) = \sum_{(l,i) \in \delta^{-}(i)} x_{li}(t_k - \tau_{li}) - \sum_{(i,j) \in \delta^{+}(i)} x_{ij}(t_k) \\
& \quad \forall i \in V, k = 1, \ldots, k_{\max} \\
& y_i(0) = y_{i,0} \quad \forall i \in V \\
& y_i(t_k) \leq a_i(t_k) \quad \forall i \in V, k = 1, \ldots, k_{\max} \\
& 0 \leq x_{ij}(t_k) \leq b_{ij}(t_k) \quad \forall (i,j) \in A, k = 0, \ldots, k_{\max} - \tau_{ij}. \quad (17b)
\end{align}
\]

In other words: The model \((17)\) computes a flow from the source \(s\) to the sink \(d\), shortly \(s-d\)-flow, with respect to some objective function \(\phi(X, Y)\). The flow decomposition (see \([1]\)) of the resulting flow can then be interpreted as routes used by evacuees to safety.

Finally, we comment on the objective functions \(\phi(X, Y)\) for the dynamic network flow model. As in Section 2, we start with a maximum flow for a given time horizon \(T\)

\[
\max \phi(X, Y) = \max \sum_{k=0}^{k_{\max}} \sum_{(i,j) \in \delta^{-}(d)} x_{ij}(t_k). \quad (18)
\]

For this objective the constraints \((17b) - (17c)\) are used and the calculation is started with an empty network, meaning \(y_{i,0} = 0\) for all \(i \in V\). The objective function for the quickest flow problem, minimizing the time horizon to clear a network, as described in literature (cf. \([18, 3, 8]\)) takes the form

\[
\min \phi(X, Y) = \min \max \left\{ 0 \leq t_k \leq \infty \mid x_{e}(t_k - \tau_e) \neq 0 \text{ for some } e \in \delta^{-}(d) \right\}. \quad (19)
\]

Again we use the constraints \((17b) - (17c)\), additionally equipped with

\[
\sum_{k=0}^{k_{\max}} \sum_{(i,d) \in A} x_{id}(t_k) = y_{s,0} \quad (20)
\]

ensuring that all people are evacuated (i.e. reach the sink \(d\)).

But for our applications, it is more interesting to deal with problems where cost terms for the propagation of hazardous material are included. To deduce objective functions similar to \((10)\) and \((11)\), we consider the unit intervals \(I_k = [t_k, t_{k+1}]\) between the time steps \(t_k\). Then, for both the static and the dynamic model, we take the mean value of the concentration parameters \(\tilde{c}_{ij}(t_k)\) and \(\overset{\cdot}{c}_i(t_k)\) for \(e = (i,j) \in A\) and \(i \in V\), respectively, over each interval \(I_k\) yielding

\[
\tilde{c}_{ij}(t_k) = \int_{t_k}^{t_{k+1}} c_e(t) \, dt, \quad \overset{\cdot}{c}_i(t_k) = \int_{t_k}^{t_{k+1}} c_i(t) \, dt \quad (21)
\]

where \(c_e(t)\) and \(c_i(t)\) are the time-dependent concentration parameters introduced in Section 2. Thus we can now reformulate the objective functions \((10)\) and \((11)\) yielding

\[
\max \left\{ \max_{(i,j) \in A, 0 \leq k \leq k_{\max}, x_{ij}(t_k) > 0} \tilde{c}_{ij}(t_k), \max_{i \in V, 0 \leq k \leq k_{\max}, y_i(t_k) > 0} \overset{\cdot}{c}_i(t_k) \right\} \quad (22)
\]
for the minimization of the maximal hazard to which the evacuees are exposed and

\[ \sum_{k=0}^{k_{\text{max}}} \left( \sum_{(i,j) \in A} \hat{c}_{ij}(t_k)x_{ij}(t_k) + \sum_{i \in V} \hat{c}_i(t_k)y_i(t_k) \right) \]  

(23)

for the minimization of the total amount of hazard to which the evacuees are exposed. Solution methods for the model (17) are explained in Section 4.2.

4. Computational results. In this section, we first present techniques to solve optimization problems described in the previous sections. We start with the numerical solution of the advection-diffusion equation (Section 4.1), which is a subproblem of the optimization problem (12) and its discrete analogon. Then we introduce some basic concepts for the solution of the dynamic network flow model (17) and sketch algorithms to solve the dynamic network flow model for the objective functions (18), (19), (22) and (23). Furthermore we present a comparison of running times in Section 4.2.2. We close this section with a numerical example comparing the quickest flow with the solutions obtained by considering the objective function (23).

4.1. Numerical solution of the advection-diffusion equation. For the numerical solution of equation (5) we use a five-point difference stencil \( \Delta h \) for the Laplacian as a discretization and discretize the auxiliary part with

\[ L_{\text{aux}} = \frac{1}{2h} \begin{pmatrix} 0 & d_x & 0 \\ -d_y & 0 & d_x \\ 0 & -d_y & 0 \end{pmatrix}. \]  

(24)

This yields a stable and second order consistent and convergent method as long as \( \kappa > h/2|d_i| \) for \( i = x, y \).

If we would now use the explicit method for the time discretization, we would run into problems with memory requirements and long running times as we would get restricted in the size of the time step because of the parabolic CFL condition

\[ \frac{\Delta t}{h^2} \leq \frac{1}{2} \]  

(25)

to ensure stability of the method. So, we use the trapezoidal rule for time discretization and obtain the two-dimensional version of the Crank-Nicolson Method

\[ \omega_{ij}^{k+1} = \omega_{ij}^k + \kappa \frac{\Delta t}{2} \left( \Delta h \omega_{ij}^k + \Delta h \omega_{ij}^k \right) - \Delta t L_{\text{aux}} \omega_{ij}^k. \]  

(26)

This method is implicit and not restricted in the step size \( \Delta t \) of the time discretization. But the Crank-Nicolson Method responds to the discontinuities in the initial conditions with oscillations (see text to equation (5)). This problem has been studied, for example, by Pearson [32] and Osterby [31]. They show that the complete annihilation of the oscillations is only possible by using the implicit method for the time discretization, which would reduce the order of accuracy from second to first. But Pearson [32] suggests a method which does not aim at a complete annihilation of the oscillations but at cumulative damping in several steps to an acceptable level. Therefore the first step is subdivided into \( M \) equal steps of length \( \Delta t = \Delta t/M \) and the resulting cumulative damping is large enough, to get rid of the oscillations. In our case using \( M = 15 \) is enough meaning not to much extra computational effort has to be spent to remove the discontinuities.
4.2. Evacuation solvers. After the distribution of the hazard has been calculated and mapped onto the network (see Section 2.1), the full network flow model with all parameters is at hand and needs to be solved. Solution methods for the several objective functions mentioned before will now be presented where we start with the description of the general concept of the time-expanded network (Section 4.2.1), which is similar to the concept of a time grid for partial differential equations. Then, we show how a maximal flow problem is solved using the time-expanded network (Section 4.2.2) and compare the running time of this solution method with the running time of the MIP (14). We go on with the quickest flow problem (Section 4.2.3) and the inclusion of the hazard in the model (Section 4.2.4). The section is closed with the discussion of a realistic example where we compare the aforementioned objectives and show their advantages and disadvantages (Section 4.2.5).

4.2.1. The time-expanded network. Common optimization algorithms for network flow problems such as the maximum flow algorithm of Ford and Fulkerson or the successive shortest path algorithm operate on static networks which do not include a time component. In the discrete problem (17), a time component is included in the network. This can be modeled explicitly by considering a copy of the network at every possible time step. We map the dynamic network onto a static time-expanded network (TEN) [1] where a feasible static flow corresponds to a feasible dynamic flow in the dynamic network. For simplicity of notation, we omit the subscript \( k \) of the time step \( t_k \) in the following and simply write \( T \) instead of \( t_{k_{max}} \). For the right time discretization we refer to Section 3.2.

**Definition 4.1 (Time-Expanded Network).** Let \( \mathcal{N} = (V, A, b, \tau) \) be a dynamic network. The Time-Expanded Network (TEN) of \( \mathcal{N} \) with time horizon \( T \) is defined as \( \mathcal{N}^T = (V^T, A^T \cup W^T) \) where

\[
V^T = \{i_t| i \in V, t \in \{0, \ldots, T\}\} \tag{27a}
\]
\[
A^T = \{(i_t, j_{t+\tau_e})| e = (i, j) \in A, t \in \{0, \ldots, T - \tau_e\}\} \tag{27b}
\]
\[
W^T = \{(i_t, i_{t+1})| i \in V, t \in \{0, \ldots, T - 1\}\} . \tag{27c}
\]

The capacities \( b^T_e \) of the arcs \( e \in A^T \cup W^T \) are chosen as

\[
b^T_e = b_{ij} \ \forall \ e = (i_t, j_{t+\tau_e}) \in A^T \tag{28a}
\]
\[
b^T_e = a_i \ \forall \ e = (i_t, i_{t+1}) \in W^T . \tag{28b}
\]

The time-expanded network contains \( T + 1 \) copies of every vertex \( i \in V \) of the original network \( \mathcal{N} \) which represent this vertex at all time steps \( t \in \{0, \ldots, T\} \). The time copies are connected in such a way that flow which enters an arc \( e \in A \) at time \( t \) arrives at the end-vertex of this arc exactly at time \( t + \tau_e \) (arcs in \( A^T \)). The arcs in \( W^T \) are called holdover arcs and model the possibility to wait at a vertex for one time step. The resulting network has no time component and algorithms for static networks can be applied. The static flow conservation constraints in the TEN correspond to the dynamic flow conservation constraints of the problem we expanded. Hence, we can interpret a static flow \( f^s \) in \( \mathcal{N}^T \) as a flow over time \((X, Y)\) in the dynamic network \( \mathcal{N} \) by setting \( x_{ij}(t) = f^s_{(i_t, j_{t+\tau_e})} \) and \( y_i(t) = f^s_{(i_t, i_{t+1})} \). Since the time-expanded network considers every time step explicitly it is straight forward to include time dependent parameters on the arcs.
Figure 1. Time-expanded network of a small network

**Example 1.** Figure 1 shows an example for a time-expanded network. In a) a dynamic network is given with parameters \((\tau_e, b_e)\) on the arcs. b) Shows the time-expanded network of a) with time horizon \(T = 5\), where the values on the arcs show their capacities. The capacity of the storage arcs is chosen as \(\infty\) in this example and is not depicted on the arcs. The dashed arcs and vertices are not necessary in the time expansion when considering flow from \(v_1\) to \(v_3\). No flow starting at vertex \(v_1\) which arrives at time \(T\) or earlier at vertex \(v_3\) can use any of these arcs and vertices.

The drawback of this general method is the size of the time-expanded network. The number of vertices and arcs of the network scale with \(T\). So, although time expansion is a useful tool, using static network flow algorithms on the time-expanded network will not give polynomial-time algorithms for the dynamic network flow problems. Note that for capacities and travel times that are constant over time there exist also polynomial time algorithms for the maximum dynamic flow problem (18) and the quickest flow problem (19) (see [37, 3]).

We now use the concept of the time-expanded network to give algorithms for the solution of the optimization problems mentioned in Section 3.

**4.2.2. An algorithm for the maximum dynamic flow problem.** To solve the maximum dynamic flow problem we have to find a dynamic \(s-d\)-flow which maximizes the flow arriving at \(d\) in a given time horizon \(T\). Since the time at which the flow starts in the source and the time at which it arrives at the sink are not important – as long as it is within the time horizon \(T\) – for the objective function, we add a super source \(S\) which is connected to every time copy of \(s\) with infinite capacity, and a super sink \(D\) which is connected to every time copy of the sink \(d\) with infinite capacity. So we reduce the problem to a single source single sink maximum flow problem in the time-expanded network which can be solved efficiently [1]. This solution is then mapped to a flow over time which is a maximum dynamic flow.

**Comparison of CPU times.** As the maximum flow problem is on the one hand a basic problem and on the other hand easy to implement for both the discrete model (17) and the MIP (14) (which reduces to a LP in case of the maximum flow problem), we use it to compare the CPU times of both models. For the analysis of the running time, depending on the size of a graph, we use a network shown in Figure 2 to which we refer as the chessboard. It consists of a square of vertices, in
which vertices are connected from bottom to top and from left to right. Furthermore to the lower left corner an arc from the source and to the upper right corner an heading to the sink are added. We choose this kind of a network as it can easily be implemented and scaled in a straightforward way.

Figure 2. The chessboard of size \( g = 5 \) has \( m = 42 \) arcs and \( n = 27 \) vertices. The numbering of both vertices and arcs goes from left to right and from bottom to top. For each arc its number is given in the graph.

We start the analysis with a graph of five vertices in a square – not counting the source and the sink. The capacities of inflow and outflow arc are \( \mu_1 = \mu_{42} = 40 \), their travel times \( \tau_1 = \tau_{42} = 1 \). The capacities and travel times of all inner arcs, i.e. all other arcs, are each chosen randomly between one and four from an equal distribution. The travel times are kept constant over time but the capacities are chosen for each time step up to the time horizon \( T = 160 \). As we choose the parameters of the inner arcs randomly and the running time may be affected by that choice we throw the die ten times. In that way we generate ten graphs with \( n = 27 \) vertices and \( m = 42 \) arcs with different parameters, which we call \( G_1^5, G_2^5, \ldots, G_{10}^5 \) – the subscript indicating the size of the square and superscript the number of the die roll. We extend these graphs by adding five vertices to the upper and the right end of the square to end up with a square of ten times ten vertices. We only choose the parameters of the inner arcs which were no element of the corresponding \( G_i^5 \) randomly between one and four, the other parameters are taken from \( G_i^5 \).

We repeat this procedure until we reach a size of 50 for the chessboard yielding the following chain of sub- and supergraphs

\[
G_5^i \subset G_{10}^i \subset \cdots \subset G_{50}^i \quad i = 1 \ldots 10.
\]

The number of vertices \( n \) and the number of arcs \( m \) for each graph \( G_g^i \) is calculated for all \( i \) by

\[
n = g^2 + 2 \quad \text{ (30)}
\]

\[
m = 2(g - 1)g + 2 = 2(n - g - 1) \quad \text{ (31)}
\]

Thus, the largest network has \( n = 2502 \) vertices and \( m = 4902 \) arcs. For all graphs \( G_g^i \) we compute the optimal solutions for both the discrete model using the time-expanded network (TEN) and the PDE-model being simplified as a LP shown in equation (14). The TEN is solved using the Boost-library [36] whereas the LP is solved using the commercial solver of CPLEX [39].

The optimal objective values of the solutions gained by the TEN and LP coincide for each graph \( G_g^i \), so we only present the mean solution times (CPU times of the solvers) in Table 1 and Figure 3, where we take the mean over the different die rolls.
Table 1. Average values of solution times for graphs of equal size for TEN (discrete model) and LP (PDE-model)

\[
\begin{array}{|c|c|c|c|c|c|c|}
\hline
\text{graph} & \#\text{vertices} & \#\text{arcs} & \#\text{variables} & \text{solution time} \text{ [s]} \\
\hline
G_5 & 27 & 42 & 11040 & 26880 & 0.40 & 2.76 \\
G_{10} & 102 & 182 & 45440 & 116480 & 1.30 & 80.26 \\
G_{15} & 227 & 422 & 103840 & 270080 & 3.20 & 487.28 \\
G_{20} & 402 & 762 & 186240 & 487680 & 5.60 & 1285.92 \\
G_{25} & 627 & 1202 & 292640 & 769280 & 8.40 & 1893.61 \\
G_{30} & 902 & 1742 & 423040 & 1114880 & 12.00 & 1878.64 \\
G_{35} & 1227 & 2382 & 577440 & 1524480 & 13.40 & 1627.69 \\
G_{40} & 1602 & 3122 & 755840 & 1998080 & 13.30 & - \\
G_{45} & 2027 & 3962 & 958240 & 2535680 & 17.30 & - \\
G_{50} & 2502 & 4902 & 1184640 & 3137280 & 14.00 & - \\
\hline
\end{array}
\]

We see that the LP is not capable to process the graphs of sizes \( g > 35 \), which is due to excessive amount of variables and memory for these graphs. For the largest graphs which we can handle using the LP we need more than 1.5 million variables and the input file for CPLEX [39] needs more than 2 GB memory. Moreover we see that the TEN algorithm is much faster (up to 250 times) computed than the solution of the LP. From Figure 3 we also see that the average solution times of both the LP and the TEN grow more slowly for greater networks. We suppose that this is due to the relatively small time horizon and enhanced preprocessing capabilities of CPLEX [39]. Since for the large networks the time horizon is not large enough for all \( s-d \)-paths to be used, less flow is shipped through the network and, thus, more variables are zero, which speeds up the computation.

Figure 3. Average values of solution time against size \( g \) of the Graphs \( G_g \) for TEN (dashed) and LP (solid). Be aware of the logarithmic scale of the figure.
4.2.3. *An algorithm for the quickest flow problem.* After having seen how well the concept of the time-expanded network worked, we now show how to use it for the solution of the quickest flow problem. Therefore we add again a super source $S$ and super sink $D$ as in the maximum dynamic flow problem. In this scenario, however, the time at which flow arrives at the sink is important. Hence, we assign *turnstile costs* [4] to the arcs that lead from any time copy of the sink to the super sink (cf. Figure 4). These costs represent the time at which flow using this arc arrives at the sink. All other arcs get a cost value of zero. Calculating a minimum cost flow with value $y_{s,0}$ and with respect to the turnstile costs in the TEN then gives a quickest flow, since if flow could arrive earlier in the network this would reduce the costs of the flow in the time-expanded network.

![Figure 4. Turnstile costs for a quickest flow computation for the graph of Figure 1 (a).](image)

**Example 2.** Figure 4 shows the network for a quickest flow calculation in the time-expanded network of Figure 1. The capacities are omitted for clarity. A super source $S$ and a super sink $D$ are connected to the network with arcs of infinite capacity. The values shown on the arcs leading to the super sink are the turnstile costs which represent the time flow arrives at the sink. The costs of the other arcs are set to zero and are not shown in the figure. Calculating a minimum cost flow in this network gives a quickest flow.

4.2.4. *An algorithm including the hazard costs.* We close the description of the discrete algorithms by showing how to include the cost function for the hazard we got in Section 3 in the optimization problem. Therefore we map the (possibly time dependent) cost function $\tilde{c}$ onto the time-expanded network by setting

$$\tilde{c}_e = \tilde{c}_{i,j}(t) \quad \forall e = (i_t, j_{t+\tau_{i,j}}) \in A^T$$

$$\tilde{c}_e = \tilde{c}_{i}(t+1) \quad \forall e = (i_t, i_{t+1}) \in W^T.$$ 

We reduce the problem again to a single source single sink problem by adding super source and super sink. The costs of the arcs which connect these two new vertices to the remaining network are set to zero. Since waiting in the source is also dangerous we have to ensure that all the flow enters the network at time $t = 0$. This can be done by connecting the super source only to the time copy of the source at $t = 0$. 

Since the sink is considered to be a save vertex this restriction is not necessary for the super sink which is still connected to all time copies of the sink.

To minimize the sum of the costs (23) it is now sufficient to calculate a minimum cost flow with respect to these costs. To minimize the maximum cost value (22) we can iteratively delete the arcs with the highest costs and check if there is still a feasible solution which can send all $y_{s,0}$ units of flow to the sink.

4.2.5. Realistic case studies. Finally, we present a numerical example for the coupled model. Therefore we consider the serial computation (see Algorithm 1) of the transportation of the hazardous material, its mapping onto the network and the calculation of the evacuation plan. In this way, the calculation of the dynamics of the hazard can be seen as a preprocessing step for the evacuation. Since we have shown in Section 4.2.2 that the calculation of the dynamic network flow model is much faster than the calculation of the MIP, we only consider the dynamic network flow model and the corresponding objective functions in this example.

**Algorithm 1** Solution algorithm

1: Transport of hazardous material as preprocessing
2: Mapping on network
3: Start evacuation solver

We start with the description of the network and its parameters. After that, we calculate the spreading of the hazard for different directions and speeds of wind. For each of these settings we then show snapshots of the resulting optimal evacuation process.

![Network for the example](image)

**Figure 5.** Network for the example. The squares and dots and lines represent the vertices and arcs of the graph, respectively. The circle indicates the source of the hazard and the squares the safe destinations.

The source of the hazard is supposed to be located in the middle of the graph (see circle in Figure 5). We consider all vertices at the boundary, which are shown as squares in Figure 5, as safe destinations, whereas the inner vertices of the graph (dots in Figure 5) are considered to be possible evacuation sources. But we restrict ourselves to only one evacuation source. The lines in Figure 5 show the arcs of the graph, where we assume that the safe destinations have only incoming arcs and all
other lines represent arcs in both directions. This is motivated by the fact that it does not make any sense to drive back into the danger zone after having reached safety. As we want to be as realistic as possible the network presented above mimics a region of about 50 km² in Southern Germany.

In order to calculate the capacities and travel times of the arcs as dimensionless numbers we fix the length of one time step $\theta$ to be one minute. For each arc we assume a constant average velocity $v_e$ depending on the category of the street considered. Depending on the velocity the cars are assumed to hold an appropriate distance $d_e$ to the one driving in front. By appropriate we understand approximately the well-known rule of thumb “halftimes speedometer”. Furthermore we assume that each car takes three persons to safety ($n_{pc} = 3$). Thus we may calculate the capacity of each arc $e = (i,j)$ by

$$b_{ij} = \frac{v_e}{d_e} n_{pc} n_{e,l} \theta,$$

where $n_{e,l}$ is the number of lanes of the street depending on its category.

The travel time $\tau_{ij}$ of each arc $e = (i,j)$ is calculated by

$$\tau_{ij} = \frac{L_e}{v_e},$$

where $L_e$ is the distance between the vertices linked by the arc.

Inserting the values of Table 2 enables us to determine the capacities of the arcs. Since the solver used cannot handle real numbers for the parameters all parameter values – capacities and travel times – are rounded to integers.

After having set up the network we now look at the solutions of the evacuation model which are gained by the time-expanded network algorithm. Since we only want to consider a single source problem, we do not evacuate all vertices. We concentrate on the vertex east of the source of the hazard, which is located nearest to it. Furthermore we give one example for the evacuation of the vertex southwest of the source of the hazard, which is well connected. We consider two directions of wind. On the one hand we consider a wind blowing east cutting the network in a northern and a southern part. On the other hand, we consider wind in southwest direction, because this region is the most densely populated. Additionally, we consider a calm, i.e. only diffusion. The initial area $\Omega_0$ of the cloud is always $4 km^2$ large and centered at the source of the hazard.

Independently of the direction of the wind we can calculate the quickest flow objective (19) for the evacuation of the vertex east of the source of the hazard ($y_{s,0} = 8787$). But from Figure 6 we see that this is – despite of being the fastest solution (cf. Table 3) – not a good solution, especially if the wind is blowing east, since the arcs are mostly covered by the hazardous cloud.
Thus we use the sum objective (23) minimizing the accumulated sum of hazard to which the evacuees are exposed. This is also a reasonable objective function from the medical point of view if an exposure to radiation is considered, since even an exposure for a duration of several hours can be treated as an one-time one [33]. We use the dynamic model (see Section 2.1) to map the cloud onto the network since this is the most realistic one. This can be seen in Figure 7 and in the comparison of the solutions in Figure 8.

Furthermore, we observe that using the static model it is possible that the evacuees head in direction of the centre of the hazardous cloud, which is intuitively no good idea. Using the dynamic model such a behaviour of the solution does not occur (see Figure 8 left picture, arc heading north).

We close this section with a comparison of the optimal solutions of the evacuation model with the sum objective for the different wind directions.
We see from Figures 9 and 10 that all arcs leaving the vertex east of the source are usable for the evacuation in the case of wind in southwestern direction and a lull and that the evacuation time is within a 20 % margin of the time of the quickest flow (cf. Table 3). The accumulated concentration for the diffusion case is higher since the center of the hazard does not move away from the vertex, where the evacuees have to wait as the capacities of the streets are not large enough. Thus a lot of people are affected by the hazard before they can be evacuated. For the wind blowing east the evacuation time doubles (cf. Table 3) as at first only the arc heading south is available (see Figure 11 left picture) and only after the cloud has passed the vertex and decayed, the arc heading north can be used. However, the arc heading east can never be used since it leads directly into the hazard.

**Outlook.** This article presents a first, rather simple, attempt to incorporate the spread of a hazardous material into an egress model for evacuation of regions based on street networks. With the definition of suitable objective functions, the best possible evacuation routes have been found, either to fastest egress or to safest...
Table 3. Comparison of evacuation times and accumulated concentration for different wind directions and strengths.

| problem         | accumulated concentration [a.u.] | evacuation time [min] |
|-----------------|----------------------------------|-----------------------|
| quickest flow   | -                                | 49                    |
| no wind         | 434298                           | 53                    |
| wind east       | 373628                           | 127                   |
| wind southwest  | 108018                           | 51                    |

evacuation. Furthermore, we have demonstrated the effectiveness of using dynamic network flows to compute solutions of the original continuous evacuation problem.

Future research will focus on extending the current models to multiple sources and to take traffic jams (similar to [6, 21]) and selfishness of evacuees into account. This will make the evacuation model nonlinear and requires adapted dynamic flow algorithms. Moreover, the model for the propagation of hazardous material might be more sophisticated. Think of time and space-dependent wind fields or even Navier-Stokes equations. Finally, instead of considering single objective functions, a multicriteria model in which two or more objectives are considered simultaneously might allow for adequate compromise solutions and form the basics of a decision support system.

Acknowledgments. M. Thiemann and S. Ruzika are partially supported by BMBF, Project REPKA, FKZ 13N9961 (TU KL).

REFERENCES

[1] R. K. Ahuja, T. L. Mananti and J. B. Orlin, “Network Flows: Theory, Algorithms, and Applications,” Prentice Hall, Englewood Cliffs, NJ, 1993.
[2] D. Armbruster, P. Degond and C. Ringhofer, A model for the dynamics of large queuing networks and supply chains, SIAM J. on Applied Mathematics, 66 (2006), 896–920.
[3] R. E. Burkard, K. Dlaska and B. Klinz, The quickest flow problem, Z. Oper. Research, 37 (1993), 31–58.
[4] L. G. Chalmet, R. L. Francis and P. B. Saunders, Network models for building evacuation, Fire Technology, 18 (1982), 90–113.
[5] W. Choi, H. W. Hamacher and S. Tufekci, Modeling of building evacuation problems by network flows with side constraints, European Journal of Operational Research, 35 (1988), 98–110.
[6] G. M. Coclite, M. Garavello and B. Piccoli, Traffic flow on a road network, SIAM Journal on Mathematical Analysis, 36 (2005), 1862–1886.
[7] C. D’Apice, S. Göttlich, M. Herty and B. Piccoli, “Modeling, Simulation and Optimization of Supply Chains: A Continuous Approach,” SIAM, Philadelphia, PA, 2010.
[8] L. Fleischer and É. Tardos, Efficient continuous-time dynamic network flow algorithms, Operations Research Letters, 23 (1998), 71–80.
[9] L. R. Ford and D. R. Fulkerson, Constructing maximal dynamic flows from static flows, Operations Research, 6 (1958), 419–433.
[10] A. Fügenschuh, S. Göttlich, M. Herty, A. Klar and A. Martin, A discrete optimization approach to large scale supply networks based on partial differential equations, SIAM Journal on Scientific Computing, 30 (2008), 1490–1507.
[11] A. Fügenschuh, M. Herty, A. Klar and A. Martin, Combinatorial and continuous models for the optimization of traffic flows on networks, SIAM Journal on Optimization, 16 (2006), 1155–1176.
[12] S. Göttlich, M. Herty and A. Klar, Network models for supply chains, Communications in Mathematical Sciences, 3 (2005), 545–559.
[13] E. Graat, C. Midden and P. Bockholts, *Complex evacuation; effects of motivation level and slope of stairs on emergency egress time in a sports stadium*, Safety Science, 31 (1999), 127–141.

[14] S. Gwynne, E. R. Galea, M. Owen, P. J. Lawrence and L. Filippidis, *A review of the methodologies used in evacuation modelling*, Fire and Materials, 23 (1999), 383–388.

[15] H. W. Hamacher, S. Heller, G. Köster and W. Klein, *A Sandwich Approach for Evacuation Time Bounds*, in “Pedestrian and Evacuation Dynamics” (eds. R.D. Peacock, E.D. Kuligowski, and J.D. Averill), Springer US, (2011), 503–514.

[16] H. W. Hamacher, K. Leiner and S. Ruzika, *Quickest Cluster Flow Problems*, in “Pedestrian and Evacuation Dynamics” (eds. R.D. Peacock, E.D. Kuligowski, and J.D. Averill), Springer US, (2011), 327–336.

[17] H. W. Hamacher and S. A. Tjandra, *Earliest arrival flows with time dependent capacity for solving evacuation problems*, in “Pedestrian and Evacuation Dynamics” (eds. M. Schreckenberger and S.D. Sharma), Springer, Berlin, (2002), 267–276.

[18] H. W. Hamacher and S. A. Tjandra, *Mathematical modelling of evacuation problems—a state of the art*, in “Pedestrian and Evacuation Dynamics” (eds. M. Schreckenberger and S.D. Sharma), Springer, Berlin, (2002), 227–266.

[19] D. Helbing, *A mathematical model for the behavior of pedestrians*, Behavioral Science, 36 (1991), 298–310.

[20] M. Herty and A. Klar, *Modeling, simulation, and optimization of traffic flow networks*, SIAM Journal on Scientific Computing, 25 (2003), 1066–1087.

[21] H. Holden and N. H. Risebro, *A mathematical model of traffic flow on a network of unidirectional roads*, SIAM Journal on Mathematical Analysis, 26 (1995), 999–1017.

[22] S. P. Hoogendoorn and P. H. L. Bovy, *Gas-kinetic modeling and simulation of pedestrian flows*, Transportation Research Record, (2000), 28–36.

[23] R. Hughes, *A continuum theory for the flow of pedestrians*, Transportation Research Part B, 36 (2002), 507–535.

[24] C. Kirchner, M. Herty, S. Göttlich and A. Klar, *Optimal control for continuous supply network models*, Networks Heterogenous Media, 1 (2006), 675–688.

[25] H. Klüpfel, M. Schreckenberg and T. Meyer-König, *Models for Crowd Movement and Egress Simulation,* Traffic and Granular Flow ’03, (2005), 357–372.

[26] A. Kneidl, M. Thiemann, A. Borrmann, S. Ruzika, H. W. Hamacher, G. Köster and E. Rank, *Bidirectional Coupling of Macroscopic and Microscopic Approaches for Pedestrian Behavior Prediction*, in “Pedestrian and Evacuation Dynamics” (eds. R.D. Peacock, E.D. Kuligowski, and J.D. Averill), Springer US, (2011), 459–470.

[27] E. Köhler, K. Langkau and M. Skutella, “Time-Expanded Graphs for Flow-Dependent Transit Times,” Lecture Notes in Computer Science, 2461 Springer, Berlin, (2002), 599–611.

[28] E. Köhler and M. Skutella, *Flows over time with load-dependent transit times*, SIAM Journal on Optimization, 15 (2005), 1185–1202.

[29] C. D. Laird, L. T. Biegler and B. G. van Bloemen Waanders, *Real-time, large-scale optimization of water network systems using a subdomain approach*, in “Real-Time PDE-Constrained Optimization” (eds. L.T. Biegler, O. Ghattas, M. Heinkenschloss, D. Keyes and B. G. van Bloemen Waanders), SIAM Series in Computational Science and Engineering, Philadelphia, PA, (2007), 289–306.

[30] C. D. Laird, L. T. Biegler, B. G. van Bloemen Waanders and R. A. Bartlett, *Contaminant source determination for water networks*, Journal of Water Resources Planning and Management, 131 (2005), 125–134.

[31] O. Österby, *Five ways of reducing the Crank-Nicolson oscillations*, BIT Numerical Mathematics, 43 (2003), 811–822.

[32] C. E. Pearson, *Impulsive end condition for diffusion equation*, Mathematics of Computation, 19 (1965), 570–576.

[33] B. Rajewsky, “Strahlendosis und Strahlenwirkung,” Thieme, Stuttgart, 1954.

[34] G. Santos and B. Aguirre, “A Critical Review of Emergency Evacuation Simulation,” Proceedings of Building Occupant Movement during Fire Emergencies, June 10-11, NIST/BFRL Publications Online, Gaithersburg, (2004), 27–52.

[35] A. Schadschneider, W. Kilingsch, H. Klüpfel, T. Kretz, C. Rogsch and A. Seyfried, *Evacuation dynamics: Empirical results, modeling and applications*, in “Encyclopedia of Complexity and System Science” (ed. B. Meyers), Springer, New York, (2009), 3142–3176.
[36] J. G. Siek, L.-Q. Lee and A. Lumsdaine, “The Boost Graph Library: User Guide and Reference Manual (C++ In-Depth Series),” Addison-Wesley, Boston, 2001.

[37] M. Skutella, “An Introduction to Network Flows Over Time,” Research Trends in Combinatorial Optimization, Springer, Berlin, (2009), 451–482.

[38] F. Southworth, “Regional Evacuation Modeling: A State-of-the-Art Review,” ORNL/TAM-11740, Oak Ridge National Laboratory, Energy Division, Oak Ridge, 1991.

[39] IBM ILOG CPLEX Optimization Studio, Cplex version 12, 2010.

Received December 2010; revised July 2011.

E-mail address: goettlich@uni-mannheim.de
E-mail address: skuehn@mathematik.uni-kl.de
E-mail address: ohst@mathematik.uni-kl.de
E-mail address: ruzika@mathematik.uni-kl.de
E-mail address: thiemann@mathematik.uni-kl.de