The Building Evacuation Problem with Shared Information

Lichun Chen, Elise Miller-Hooks

Department of Civil and Environmental Engineering, University of Maryland, College Park, Maryland 20742

Received 30 April 2007; revised 23 October 2007; accepted 15 February 2008
DOI 10.1002/nav.20288
Published online 28 March 2008 in Wiley InterScience (www.interscience.wiley.com).

Abstract: In this article, the Building Evacuation Problem with Shared Information (BEPSI) is formulated as a mixed integer linear program, where the objective is to determine the set of routes along which to send evacuees (supply) from multiple locations throughout a building (sources) to the exits (sinks) such that the total time until all evacuees reach the exits is minimized. The formulation explicitly incorporates the constraints of shared information in providing online instructions to evacuees, ensuring that evacuees departing from an intermediate or source location at a mutual point in time receive common instructions. Arc travel time and capacity, as well as supply at the nodes, are permitted to vary with time and capacity is assumed to be recaptured over time. The BEPSI is shown to be NP-hard. An exact technique based on Benders decomposition is proposed for its solution. Computational results from numerical experiments on a real-world network representing a four-story building are given. Results of experiments employing Benders cuts generated in solving a given problem instance as initial cuts in addressing an updated problem instance are also provided. © 2008 Wiley Periodicals, Inc. Naval Research Logistics 55: 363–376, 2008

Keywords: evacuation; shared information; time-varying, dynamic networks; benders decomposition; unsplittable flows; reoptimization

1. INTRODUCTION

The Building Evacuation Problem with Shared Information (BEPSI) is addressed in this article. The objective of the BEPSI is to determine a set of evacuation routes and the assignment of evacuees to these routes for a large burning building or a building that has come under attack by enemy or natural catastrophe such that the total evacuation time is minimized. The term building is used generically throughout this work and refers to any structure that houses people and other assets, such as a high-rise residential building, a military complex like the Pentagon, or a large ship. Resulting routes could be updated in response to new information ascertained about the operational capacity of the building's circulation systems (i.e. the means of egress). Such routes and updates to these routes during the course of the evacuation could be provided in the form of instructions to the evacuees via changeable message signs, photoluminescent signage, voice evacuation systems, or other technologies that would support real-time public information updates in substandard conditions. Thus, any instructions that are provided at a particular location in the building will likely be simultaneously received by many evacuees. That is, evacuees departing from an intermediate or source location at a particular point in time receive common instructions as to how to proceed (i.e. shared information). The potential for providing such updated evacuation instructions given real-time information and predictions of the condition of the building's structures and circulation systems based on data from sensor systems is described in [24]. Existing optimization approaches in the literature cannot guarantee that common instructions will be generated at intermediate locations at any given point in time.

Typical building evacuation plans are developed predisaster for no specific threat and these plans are posted throughout the building. Such plans could, in an actual evacuation, route evacuees into harms way (e.g. to a stairwell with untenable conditions), leaving evacuees to their own devices to find alternative (safer) routes. Past experience has demonstrated that two main hindrances to the movement of evacuees in a building evacuation exist: (1) inappropriate selection of escape pathways and (2) congestion along the safest pathways [20]. Instructions generated for the specific circumstances leading to the need for the evacuation can lead to significant improvements in escape pathway selection. Moreover, explicit consideration of the number of people that such pathways can support in developing real-time evacuation instructions can lead to reduced congestion throughout the building and greater likelihood of successful egress.

Correspondence to: E. Miller-Hooks (elisemh@umd.edu)

In this article, the BEPSI is formulated as a mixed integer linear program, where the objective is to determine the set of routes along which to send evacuees (supply) from multiple locations throughout the building (sources) to the building exits (sinks) such that the total time required of all evacuees to reach the exits is minimized. The formulation explicitly incorporates the constraints of shared information; thus, feasible solutions must not contain more than one path from a node at a given departure time. Arc travel time and capacity, as well as supply at the nodes, are permitted to vary with time (i.e. the network is permitted to be time-varying) and capacity is assumed to be recaptured over time (i.e. the network is dynamic). Thus, the formulation can be viewed as a time-dependent, dynamic transshipment problem with side constraints. A similar distinction between time-dependence and problem dynamics is made in [25]. An exact solution technique based on Benders decomposition is proposed for solution of the BEPSI.

Optimization techniques have been proposed for use in determining optimal evacuation routes for both building and regional evacuation over the past few decades and a number of these works develop network flow-based solution techniques that consider the dynamic and, in some cases, the time-dependent network properties. See [11] and [25] for a review of relevant works in the literature. Additional relevant works published in the past couple of years include [1, 16, 21, 23]. All of these works assume that when two or more units of flow (i.e. the evacuees) arrive at an intermediate node, instructions can be provided that permit the flow to split among various routes. Thus, the instructions may, for example, send a subset of flow units along one route and the remaining units along another route. The provision of such instructions that require evacuees to separate at intermediate locations despite that they have arrived at this location together would not likely be palatable and could lead to confusion, or worse, chaos.

To corroborate this concept of a need for shared instructions, research has shown that in a crisis, such as would arise in an evacuation, people look to each other for cues in making decisions as to how to proceed [12, 15]. Helbing et al. [12], for example, noted a strong tendency towards collective behavior, where people follow the actions of others in evacuations involving crowds. An emergent norm that guides the group's behavior forms as people seek coordinated, collective action [30]. In addition, Sime [28] stated that during a fire, people will gravitate to familiar people and if groups are split, they seek to reunite during the evacuation. Wenger and Vigo [31] postulated that preexisting and emergent social relationships impact collective behavior. Observations from these works support the need for providing instructions that do not require a group of evacuees arriving at an intermediate location to split apart, i.e. that support a group's desire for collective action.

A similar concept of "unsplittable flow" has been employed in formulating bin-packing, virtual-circuit routing, scheduling, and load balancing problems (see, for example, [3, 8, 18]). The unsplittable flow problem seeks to route numerous commodities each along a single route from a source to a desired sink while respecting arc capacity limitations. In the limit, if only one commodity is considered, this problem would be identical to a static version of the BEPSI with one sink and supply at only one origin. Of greater relevance, perhaps, is work by Lu et al [21]. Their work proposed a heuristic for evacuating all evacuees who begin at a particular source node along a single route such that arc capacity limitations are respected. Multiple sources are considered. If such routes cross (i.e. are not independent), such a solution could require evacuees simultaneously arriving at an intermediate node from different origins to take different routes out of that intermediate node.

In the next section, a mathematical formulation is proposed for the BEPSI that explicitly considers the inherent dynamic and time-varying nature of the evacuation problem. By explicitly considering these characteristics, resulting solutions will avoid sending evacuees to corridors or stairwells when conditions at these locations are expected to be untenable or difficult to traverse. The authors know of no works in the literature that address the issue of shared information that arises in this building evacuation problem. In addition, in the next section, the BEPSI is shown to be NP-hard. In Section 3, a Benders decomposition approach for solving the BEPSI is proposed and is illustrated on an example 5-node network. Computational results from numerical experiments on a real-world network representing a four-story building are given in Section 4. Conclusions and directions for future work are discussed in Section 5.

2. THE EVACUATION PROBLEM WITH SHARED INFORMATION

The evacuation problem with shared information exploits a network representation of a building. In such a representation, the network represents the layout of the circulation systems of the building, where nodes correspond with locations inside the building (such as offices, meeting rooms, lobbies, lavatories, building exits, and corridor intersections) and arcs correspond with the passageways that connect these locations (such as staircases, elevator shafts, doorways, corridors, and ramps). A cost is often associated with the use of an arc. In evacuation problems, the cost is typically given in terms of the time it takes to traverse the arc, known as the arc traversal time. When large numbers of people must be simultaneously evacuated, issues concerning capacity of the network arcs arise. The capacity of an arc is the number of people that can pass through the associated passageway per unit of time. The arc capacities are dependent upon the size and type of passageway that the arcs represent. Arc traversal times are a function of network conditions. The nodes at which the people are located when the evacuation begins are called source nodes and the exits are referred to as sink nodes.

2.1. Preliminaries

Consider a time-dependent, dynamic network represented by $\Im = (G, u, \tau)$, $G = (N, A, \{0, \dots, T\})$, where $N = \{1, \dots, n\}$ is the set of nodes, $A = \{(i, j) | i, j \in N\}$ is the set of directed arcs, and T is the analysis period of interest discretized into small time intervals $\{0, \dots, T\}$. It is assumed that all evacuees can egress before time T; although, one can set a tighter bound on the evacuation time. Note that T may be an expert-generated bound to model physical processes, such as the time by which conditions are expected to become untenable due to smoke or fire spread or complete collapse of the building's structures. Alternatively, T may be set simply to ascertain the number of people that will escape in a given time interval. One could also seek an optimal T, i.e. the minimum time by which every evacuee could exit the building. In this work, we focus on minimizing the total time required for all evacuees to exit, instead of minimizing T, because solutions to this latter problem can include rather poor paths for many of the evacuees. That is, there is no incentive to reduce the evacuation time of any evacuee, as long as that time is below the optimal T-bound. It is assumed that T is large enough to ensure feasibility.

Each arc $(i, j) \in A$ has associated with it a positive time-varying capacity and a nonnegative time-varying traversal time. The capacity of arc (i, j) at departure time t is denoted by $u_{ij}(t)$ with integral domain and range. Instead of representing the actual flow at any given time, the capacity of an arc is the maximum flow released on the arc at a given departure time. That is, the capacity limits the rate of flow into an arc. As flow leaves node i at some departure time t, the time it takes to reach node j, i.e. the travel time along arc (i, j), is given by positive valued $\tau_{ij}(t)$. The arc travel time is defined upon entering an arc, and is assumed to be constant for the duration of travel along that arc. To maintain generality, nonFIFO[1] (as well as FIFO) travel times are permitted. Thus, it is possible for a unit of flow to leave node i ahead of another flow unit, but arrive later than that flow unit. However, since conditions are assumed to worsen over time in circumstances warranting evacuation, an assumption of FIFO travel times may be acceptable. Fleischmann et al. [9] proposed a smooth travel time function to exclude the possibility of nonFIFO travel times, ensuring a strictly monotonically increasing arrival time function, in the context of vehicular traffic. Travel time estimates can be obtained via historical data, sensor technologies or from a function of capacity. The methodology is general enough to support all such estimation methods and resulting functions.

[1] A FIFO (First-In, First-Out) network ensures that one can never arrive earlier by departing later when traveling along the same path.

Holdover arcs $(i, i), \forall i \in N$, are introduced at the nodes to allow evacuees to arrive at intermediate locations and wait for capacity to become available on outgoing arcs. Traversal times and capacities of the holdover arcs are set to one unit and infinity, respectively, $\forall i \in N$ at each departure time $t \in \{0, \dots, T\}$. The traversal time of the holdover arc at the sink node is set to zero for all departure time intervals, because there is no penalty for arriving at the sink before T.

The number of source nodes is denoted by M and the set of source nodes and sink node are denoted by $K = \{k_1, k_2, \dots, k_M\}$ and l, respectively. The supply at any source node $k_m \in K$ at time t is denoted by $b_{k_m}(t)$ and can take on positive values for any $t \in \{0, \dots, T-1\}$. The supply of any intermediate node is assumed to be zero. Without loss of generality, it is assumed that only one sink exists. One can model additional sinks by adding a super sink to the network and connecting each actual sink to this node with arcs of zero travel time and infinite capacity. It is assumed that at $t = T$, the supply at node l will be equal to the total supply, $B = \sum_{k_i \in K} \sum_{t=1}^{T} b_{k_i}(t)$, so that $b_l(T) = -B$. This does not prevent the flow from arriving at the sink at an earlier time. When flow arrives before time T, it simply waits without penalty until time T to satisfy the demand. Supplies at transshipment nodes are zero at all times. It is assumed that the arc travel time and capacity and supply at the source nodes are known *a priori*.

2.2. Mixed integer programming formulation

The BEPSI is formulated as a mixed integer linear program. Non-negative decision variable $x_{ij}(t)$ represents the rate of flow that leaves node i at time t along arc (i, j), while binary variable $\lambda_{ij}(t)$ determines the arcs to be selected. The flow $x_{ij}(t)$ arrives at node j at time $t + \tau_{ij}(t)$. The set of arcs directed in and out of a node i are given by $\Gamma^-(i) = \{j | (j, i) \in A\}$ and $\Gamma^+(i) = \{j | (i, j) \in A\}$, respectively. The BEPSI is formulated as follows.

$$\text{P: min} \quad \sum_{(i,j) \in A} \sum_{t \in \{0,\dots,T\}} \tau_{ij}(t) x_{ij}(t) \tag{1}$$

subject to:

$$\sum_{j \in \Gamma^+(i)} x_{ij}(t) - \sum_{j \in \Gamma^-(i)} \sum_{\{\bar{t} | \bar{t} + \tau_{ji}(\bar{t}) = t\}} x_{ji}(\bar{t}) = b_i(t), \quad \forall i \in N, t \in \{0, \dots, T\} \tag{2}$$

$$\lambda_{ij}(t) \le x_{ij}(t) \le \lambda_{ij}(t) u_{ij}(t), \quad \forall (i, j) \in A, t \in \{0, \dots, T\} \tag{3}$$

$$\sum_{j\in\Gamma^+(i),j\neq i} \lambda_{ij}(t) \le 1, \quad \forall i \in N\backslash l, t \in \{0,\dots,T\} \quad (4)$$

$$x_{ij}(t) \ge 0, \lambda_{ij}(t)\text{binary}, \ \forall(i,j) \in A, t \in \{0,\dots,T\} \quad (5)$$

In this model, the objective function (1) seeks to minimize the total time to send all flow from the source nodes to the sink. The mapping $x : A \times \{0,\dots,T\} \to Z_0^+$ is said to be a feasible solution if it satisfies four sets of constraints, i.e. flow conservation constraints (2), capacity constraints (3), shared information constraints (4), and nonnegativity constraints (5). Constraints (2) were first proposed by Miller-Hooks and Stock Patterson [25] to model flow conservation constraints for the time-dependent quickest flow problem (TDQFP) (where flows are permitted to split at all nodes). Similar constraints are proposed in [29] for addressing the multisource version of the TDQFP. Constraints (3) are logical constraints that impose lower and upper bounds on the flow that can pass through each arc at a given departure time. The bounds depend on the choice of arcs that will contribute to the solution paths and aid in prohibiting splittable flows. Constraints (4) allow splittable flows if the flow is split between a single outgoing arc and the holdover arc at that node. Problem (P) can be viewed as the multisource version of the TDQFP with side constraints. Solution of the TDQFP may result in split flows at source and intermediate nodes.

2.3. Complexity

In this section, it is shown that problem (P) corresponding to the BEPSI is NP-hard.

THEOREM 1: The evacuation problem with shared information, with or without storage of flow at intermediate nodes, is NP-hard in the strong sense ($M > 1$).

PROOF: We prove this by a reduction from the three-partition problem, which is NP-complete in the strong sense [10].

Three-Partition Problem (3-Partition): Given a set of $3n$ items, $n \in Z^+$, with associated sizes $b_1,\dots,b_{3n} \in Z^+$ that satisfy $\frac{B}{4} \le b_i \le \frac{B}{2}$ and $\sum_{i=1}^{3n} b_i = nB$ for some bound $B \in Z^+$. The task is to decide whether or not the set can be partitioned into n disjoint sets $S_1, S_2,\dots,S_n$ such that for $j \in \{1,\dots,n\}$, $\sum_{i\in S_j} b_i = B$.

Given an instance of 3-Partition, a network can be constructed with multiple sources $a_1,\dots,a_{3n}$ and single sink l, as shown in Fig. 1, in polynomial time.

Supply associated with each source node a_i is b_i such that $\sum_{i=1}^{3n} b_i = nB$. Note that supply is assumed to be available at time 0. All arcs in the network have unit transit time. Without loss of generality, time bound $T := 2$.

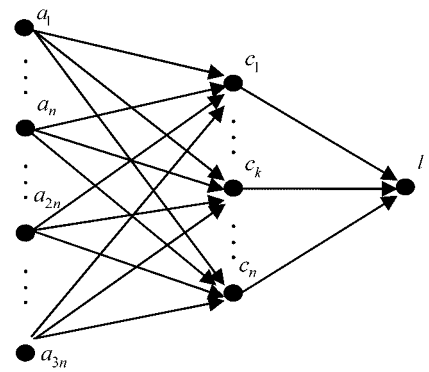


Figure 1. Reduction from 3-partition.

Arc capacities are defined by: $u(a_i, c_k) := b_i$ for $i \in \{1,\dots,3n\}$ and $k \in \{1,\dots,n\}$ and $u(c_k, l) := B$ for $k \in \{1,\dots,n\}$.

It is shown that a set of routes along which nB units of flow can be shipped from sources $a_1,\dots,a_{3n}$ to sink l within T, given that flow cannot be split at nodes $a_1,\dots,a_{3n}$, exists iff there is a yes solution to the 3-Partition instance.

If: If the underlying instance of 3-Partition is a "yes" instance, then there is a partition $S_1,\dots,S_n$ of $\{1,\dots,3n\}$ such that for $j \in \{1,\dots,n\}$, $\sum_{i\in S_j} b_i = B$. The set of routes can be generated by shipping b_i units along arc (a_i, c_k) for every $i \in S_k$. Then B units of flow will be sent on to the sink from node c_k. Thus, nB units of flow arrive at sink l at time 2.

Only if: It remains to be shown that the existence of a flow that satisfies the conditions that all units of flow leaving the same node can take only one direction and that the last unit of flow arrives at the sink no later than time T yields a feasible solution to the corresponding 3-Partition problem instance. Denote flow on any arc (a_i, c_k) at time t by $x_{(a_i,c_k)}(t)$. The binary variable $\lambda_{(a_i,c_k)}(t)$ represents that if the arc (a_i, c_k) is contained in the solution to the special instance of BEPSI problem, then

$$\sum_{i\in\{1,\dots,3n\}} x_{(a_i,c_k)}(0) = B \text{ and } x_{(c_k,l)}(1) = B, \quad \forall k \in \{1,\dots,n\}$$

$$\text{and} \sum_{k\in\{1,\dots,n\}} \lambda_{(a_i,c_k)}(0) = 1, \forall i \in \{1,\dots,3n\}.$$

It follows that

$$\sum_{i\in\{i|\lambda_{(a_i,c_k)}(0)=1\}} b_i = B, \forall k \in \{1,\dots,n\}$$

$$\therefore S_k = \{i|\lambda_{(a_i,c_k)}(0) = 1, i \in \{1,\dots,3n\}\}, \quad \forall k \in \{1,\dots,n\}$$

Hence, n sets of arcs that carry a positive amount of flow into node $c_k, \forall k \in \{1,\dots,n\}$ induce the partition of n disjoint sets satisfying $\sum_{i\in S_j} b_i = B, \forall j \in \{1,\dots,n\}$. Note that since all the arcs in the network have unit traversal time and the time bound is 2, no flow will be shipped along any holdover arc in a feasible solution of problem (P). While no holdover arcs are employed, such arcs are available, and therefore, the reduction works for both models, with and without storage. □

3. EXACT SOLUTION TECHNIQUE BASED ON BENDERS DECOMPOSITION

The formulation (P) contains a set of integer variables representing the selection of arcs, and a set of continuous variables representing the flow along each arc. The number of variables is large, even for mid-size instances; however, this structure is suitable for mathematical decomposition. An exact algorithm based on Benders decomposition to solve Problem (P), i.e. the BEPSI, is proposed herein. Benders decomposition [2] has been successfully applied to solve many mixed integer programs. See, for example, [5] and [6], both of which successfully employed Benders decomposition to solve difficult network design problems.

The original problem is reformulated using Benders decomposition into a subproblem, a pure network flow problem containing the continuous flow variables, and a master problem containing the binary arc selection variables. Benders cuts are generated by solution of the subproblem and are added to the relaxed master problem at each iteration, progressively constraining the relaxed master problem. The cuts reduce the number of flow variables that must be considered, even at the expense of increasing the number of constraints.

3.1. Benders Subproblem

Let λ be the 0-1 vector satisfying the shared information constraints (4) and let Λ be the set of valid λ. To obtain the primal subproblem, the values of λ must be fixed. For some fixed $\tilde{\lambda} \in \Lambda$ and variables $x_{ij}(t)$, the primal subproblem can be given as follows.

$$\mathrm{S_p}(\tilde{\lambda})\text{: min} \quad \sum_{(i,j)\in A} \sum_{t\in\{0,\ldots,T\}} \tau_{ij}(t)x_{ij}(t) \tag{6}$$

subject to:

$$\sum_{j\in\Gamma^+(i)} x_{ij}(t) - \sum_{j\in\Gamma^-(i)} \sum_{\{\bar{t}|\bar{t}+\tau_{ji}(\bar{t})=t\}} x_{ji}(\bar{t}) = b_i(t),$$
$$\forall i \in N, t \in \{0,\ldots,T\} \tag{7}$$

$$\tilde{\lambda}_{ij}(t) \le x_{ij}(t) \le \tilde{\lambda}_{ij}(t)u_{ij}(t),$$
$$\forall (i,j) \in A, t \in \{0,\ldots,T\} \tag{8}$$

$$x_{ij}(t) \ge 0 \quad \forall (i,j) \in A, t \in \{0,\ldots,T\} \tag{9}$$

Since $\tilde{\lambda}_{ij}(t)$ is a constant in this formulation, constraints (8) become simple lower and upper bounds on the $x_{ij}(t)$ variables. The selection of arcs is made in solving the relaxed master problem. Thus, all that remains is to determine the amount of flow to ship along these arcs. The lower bounds on $x_{ij}(t)$ variables can be dropped without impacting the optimal solution of problem (P). Because of the fact that the objective function does not contain the $\lambda_{ij}(t)$ variables, the optimal solution (λ^*, x^*) for the relaxed problem (without lower bounds) can be used to construct an optimal solution $(\hat{\lambda}^*, x^*)$ for problem (P) with the same objective function value. It was observed in preliminary experiments that computational effort is reduced by dropping the lower bounds. In addition, arc set $A = \{(i,j)|i,j \in N\}$ can be partitioned into the following three disjoint sets:

$$I_1(A) = \{(i,j)|i,j \in N \text{ and } \Gamma^+(i) \ge 2\},$$
$$I_2(A) = \{(i,j)|i,j \in N \text{ and } \Gamma^+(i) = 1\}, \text{ and}$$
$$I_3(A) = \{(i,i)|i \in N\}.$$

Thus, $A = I_1(A) \cup I_2(A) \cup I_3(A)$. The sub-problem $(\mathrm{S_p}(\tilde{\lambda}))$ can be rewritten as:

$$\mathrm{RS_p}(\tilde{\lambda})\text{: min} \quad \sum_{(i,j)\in A} \sum_{t\in\{0,\ldots,T\}} \tau_{ij}(t)x_{ij}(t) \tag{6}$$

subject to:

$$\sum_{j\in\Gamma^+(i)} x_{ij}(t) - \sum_{j\in\Gamma^-(i)} \sum_{\{\bar{t}|\bar{t}+\tau_{ji}(\bar{t})=t\}} x_{ji}(\bar{t}) = b_i(t),$$
$$\forall i \in N, t \in \{0,\ldots,T\} \tag{7}$$

$$x_{ij}(t) \le \tilde{\lambda}_{ij}(t)u_{ij}(t), \quad \forall (i,j) \in I_1(A), t \in \{0,\ldots,T\} \tag{8a}$$

$$x_{ij}(t) \le u_{ij}(t), \quad \forall (i,j) \in I_2(A), t \in \{0,\ldots,T\} \tag{8b}$$

$$x_{ij}(t) \ge 0 \quad \forall (i,j) \in A, t \in \{0,\ldots,T\} \tag{9}$$

Subproblems $(\mathrm{RS_p}(\tilde{\lambda}))$ and $(\mathrm{S_p}(\tilde{\lambda}))$ are equivalent mathematical descriptions; however, significant improvement in computational performance of the Benders decomposition approach can be obtained by using $(\mathrm{RS_p}(\tilde{\lambda}))$ in place of $(\mathrm{S_p}(\tilde{\lambda}))$. Subproblem $(\mathrm{RS_p}(\tilde{\lambda}))$ has a pure network flow structure and the constraint matrix is totally unimodular. Hence, an integral solution is guaranteed.

The dual of the primal subproblem, called the dual subproblem, is given as problem $(\mathrm{DRS_p}(\tilde{\lambda}))$ as follows.

$$\mathrm{DRS_p}(\tilde{\lambda})\text{:Max} \sum_{t\in\{0,\ldots,T\}} \left(\sum_{i\in N} \pi_i(t) b_i(t) + \sum_{(i,j)\in I_2(A)} u_{ij}(t) m_{ij}(t) + \sum_{(i,j)\in I_1(A)} \tilde{\lambda}_{ij}(t) u_{ij}(t) m_{ij}(t) \right) \tag{10}$$

subject to:

$$\pi_i(t) - \pi_j(t+\tau_{ij}(t)) + m_{ij}(t) \le \tau_{ij}(t), \quad \forall (i,j) \in A \backslash I_3(A), t \in \{0,\ldots,T\} \tag{11}$$

$$m_{ij}(t) \le 0, \quad \forall (i,j) \in A \backslash I_3(A), t \in \{0,\ldots,T\} \tag{12}$$

Here, $\pi_i(t)$ for $i \in N$ and $t \in T$ are the dual variables associated with constraints (7) and $m_{ij}(t)$ for $i \in N$ and $t \in T$ are the dual variables associated with constraints (8a) and (8b). Let D denote the polyhedron defined by constraints (11) and (12), and let P_D and R_D be the complete sets of extreme points and extreme rays of D, respectively. The null vector 0 satisfies constraints (11) and (12); thus, the dual subproblem is always feasible. By the weak duality theorem, the primal subproblem is either infeasible or feasible and bounded if the dual is feasible. To exclude the possibility of primal infeasibility, the following inequality must hold:

$$\sum_{t\in\{0,\ldots,T\}} \left(\sum_{i\in N} \pi_i(t) b_i(t) + \sum_{(i,j)\in I_2(A)} u_{ij}(t) m_{ij}(t) + \sum_{(i,j)\in I_1(A)} \tilde{\lambda}_{ij}(t) u_{ij}(t) m_{ij}(t) \right) \le 0, \quad \forall (\pi, m) \in R_D.$$

If the dual subproblem is bounded and the primal subproblem is feasible, the optimal value of both problems is given by

$$\operatorname*{Max}_{(\pi,m)\in P_D} \sum_{t\in\{0,\ldots,T\}} \left(\sum_{i\in N} \pi_i(t) b_i(t) + \sum_{(i,j)\in I_2(A)} u_{ij}(t) m_{ij}(t) + \sum_{(i,j)\in I_1(A)} \tilde{\lambda}_{ij}(t) u_{ij}(t) m_{ij}(t) \right).$$

3.2. Benders Relaxed Master Problem

The Benders master problem is obtained by replacing constraints (2), (3), and (4) by Benders cuts (14) and (15). Constraints (14) are optimality cuts that ensure corresponding nonoptimal solutions are excluded. Constraints (15) are feasibility cuts that ensure the resulting primal subproblem is feasible. Introducing the additional free variable Z, problem (P) can be reformulated as the following equivalent problem ($\bar{\mathrm{P}}$).

$$(\bar{\mathrm{P}})\text{: min} \quad Z \tag{13}$$

subject to:

$$Z - \sum_{t\in\{0,\ldots,T\}} \sum_{(i,j)\in I_1(A)} u_{ij}(t) m_{ij}(t) \lambda_{ij}(t) \ge \sum_{t\in\{0,\ldots,T\}} \left(\sum_{i\in N} \pi_i(t) b_i(t) + \sum_{(i,j)\in I_2(A)} u_{ij}(t) m_{ij}(t) \right), \quad (\pi, m) \in P_D \tag{14}$$

$$\sum_{t\in\{0,\ldots,T\}} \sum_{(i,j)\in I_1(A)} u_{ij}(t) m_{ij}(t) \lambda_{ij}(t) \le - \sum_{t\in\{0,\ldots,T\}} \left(\sum_{i\in N} \pi_i(t) b_i(t) + \sum_{(i,j)\in I_2(A)} u_{ij}(t) m_{ij}(t) \right), \quad (\pi, m) \in R_D \tag{15}$$

$$\sum_{j\in\Gamma^+(i), j\ne i} \lambda_{ij}(t) \le 1, \quad \forall i \in N \backslash l, t \in \{0,\ldots,T\} \tag{16}$$

$$\lambda_{ij}(t) \text{binary}, \quad \forall (i,j) \in A, t \in \{0,\ldots,T\} \tag{17}$$

Constraints (14) and (15) need not be enumerated exhaustively, because most of the constraints will be inactive in the optimal solution. Thus, a relaxation of problem ($\bar{\mathrm{P}}$) can be obtained by dropping constraints (14) and (15) and iteratively adding them to the relaxation until optimality is achieved. Results of preliminary experiments show that when beginning with $R_D = \emptyset$, resulting subproblems are likely to be infeasible and Benders decomposition may be very slow to converge. This concern is addressed by augmenting the relaxed master problem with valid, stronger inequalities that can reduce the number of iterations required to reach optimality.

PROPOSITION 1: In FIFO networks, if in the optimal solution to the BEPSI, flow is shipped from node $i (i \ne l)$ at time t along a holdover arc, then

$$\sum_{j\in\Gamma^+(i), j\ne i} \lambda_{ij}(t) = 1, \quad \forall t \in \{0,\ldots,T-1\}.$$

DISCUSSION: Let $\{x_{ij}(t)\}_{\forall(i,j)\in A, t\in\{0,\ldots,T-1\}}$ and $\{\lambda_{ij}(t)\}_{\forall(i,j)\in A, t\in\{0,\ldots,T-1\}}$ be the optimal solution. Suppose that in this solution, $\lambda_{ii}(t) = 1$ and $\sum_{j\in\Gamma^+(i), j\ne i} \lambda_{ij}(t) = 0$ for some node $i (i \ne l)$ at time t. Without loss of generality, suppose that $\lambda_{ij}(t+1) = 1, j \in \Gamma^+(i)$ and $j \ne i$. Then a new solution can be constructed where $\lambda_{ii}(t) = 0, \lambda_{ij}(t) = 1$, $\lambda_{jj}(t + \tau_{ij}(t)) = 1$, constraints (2)–(5) are satisfied and the objective function value is lower than in the optimal solution (because the arc traversal times cannot improve over time), contradicting the assumption that the original solution is optimal.

According to proposition 1, for any node $i (i \neq l)$ at time t, $\sum_{j \in \Gamma^+(i), j \neq i} \lambda_{ij}(t) \geq \lambda_{ii}(t)$ holds. Constraints (18a) and (18b) represent the relationship between inflow and outflow in the FIFO network (capacities are usually deteriorating in the evacuation problem), where σ is the maximum in-degree for any i in N.

$$\sigma \sum_{j \in \Gamma^+(i), j \neq i} \lambda_{ij}(t) - \sum_{j \in \Gamma^-(i)} \sum_{\{\bar{t} | \bar{t} + \tau_{ji}(\bar{t}) = t\}} \lambda_{ji}(\bar{t}) \geq 0,$$
$$\forall i \in N \backslash l, t \in \{0, \ldots, T\}, \text{ and} \tag{18a}$$
$$- \sum_{j \in \Gamma^+(i), j \neq i} \lambda_{ij}(t) + \sum_{j \in \Gamma^-(i)} \sum_{\{\bar{t} | \bar{t} + \tau_{ji}(\bar{t}) = t\}} \lambda_{ji}(\bar{t}) \geq 0,$$
$$\forall i \in N \backslash l, t \in \{0, \ldots, T\}, \tag{18b}$$

In addition, the concept of Pareto-optimal cuts is employed. Similar to other network flow problems, subproblem $(\mathrm{RS_p}(\tilde{\lambda}))$ is often degenerate and there may exist multiple optimal solutions which lead to different optimality cuts. Pareto-optimal cuts were defined as any cut that is not dominated by any other cut in [22]. By employing a Pareto-optimal cut in place of an optimality cut obtained from any optimal solution that is identified, a stronger cut may be obtained. As applied to solving sub-problem $(\mathrm{RS_p}(\tilde{\lambda}))$, the Pareto-optimal cuts can be generated by solving the following auxiliary dual sub-problem:

$$\text{Max} \sum_{t \in \{0, \ldots, T\}} \left(\sum_{i \in N} \pi_i(t) b_i(t) + \sum_{(i,j) \in I_2(A)} u_{ij}(t) m_{ij}(t) + \sum_{(i,j) \in I_1(A)} \lambda^0_{ij}(t) u_{ij}(t) m_{ij}(t) \right) \tag{19}$$

$$s.t. \sum_{t \in \{0, \ldots, T\}} \left(\sum_{i \in N} \pi_i(t) b_i(t) + \sum_{(i,j) \in I_2(A)} u_{ij}(t) m_{ij}(t) + \sum_{(i,j) \in I_1(A)} \tilde{\lambda}_{ij}(t) u_{ij}(t) m_{ij}(t) \right) = Z(\tilde{\lambda}) \tag{20}$$

$$(\pi, m) \in \Omega \tag{21}$$

where $\{\lambda^0_{ij}(t)\}$ is a core point of and $Z(\tilde{\lambda})$ is the optimal objective value of problem $(\mathrm{DRS_p}(\tilde{\lambda}))$. Constraint (20) ensures that the Pareto-optimal solution determined by solving this dual sub-problem corresponds with an alternative optimal solution to sub-problem $(\mathrm{DRS_p}(\tilde{\lambda}))$. Constraint (21) is equivalent to constraints (11) and (12).

Instead of solving the auxiliary dual problem directly, one can solve its primal problem, which is equivalent to primal sub-problem $(\mathrm{RS_p}(\tilde{\lambda}))$ with an additional variable and minor changes in the right-hand side values. This approach is due to Magnanti and Wong [22].

3.3. Benders Decomposition Algorithm

Once problem (P) has been reformulated as in Section 3.2, the BD (Benders decomposition) algorithm, proposed in this section, can be applied iteratively over the relaxed master and sub-problems until convergence. The algorithm begins by solving the relaxed master problem to determine those arcs along which flow will be shipped, i.e. the necessary input for solution of the sub-problem. Let s represent the iteration number. Let $P_{sD} \subset P_D$ represent a restricted set of extreme points and $R_{sD} \subset R_D$ a restricted set of extreme rays. Problem $(\bar{\mathrm{P}}^s)$ is obtained by replacing P_D and R_D with P_{sD} and R_{sD} in iteration s. Sets P_{sD} and R_{sD} are determined from solution of the sub-problem from iterations 1 to s. Each of these extreme points or extreme rays produces a Benders cut. These cuts are iteratively added to the relaxed master problem during the execution of the Benders decomposition algorithm.

Problem $(\bar{\mathrm{P}})$ can be relaxed further: It is not necessary to generate all constraints (16). If constraints (16) in problem $(\bar{\mathrm{P}})$ were relaxed, a subset of nodes may contain flow that splits in the optimal solution to this relaxed problem. For many problem instances, this subset is relatively small in comparison to the number of nodes. Since computational effort significantly increases with the number of constraints (16), and since many of these constraints will be inactive at optimality, those constraints that are violated in an iteration can be added to the relaxed master problem iteratively. This procedure is summarized in step 3 of the BD algorithm, which is described next.

Algorithm BD

Step 1: Set $t := 1$. Solve problem $\mathrm{RS_p}(\tilde{\lambda})$, where $\tilde{\lambda}$ is a 1's vector. Let Ω^1 be the set of nodes where flow splits.

Step 2: Set $s := 1$, $P^1_{1D} := \emptyset$, $R^1_{1D} := \emptyset$.

Step 2.1: Solve problem $(\bar{\mathrm{P}}^t_s)$. If it has no feasible solution, stop; otherwise, let λ^t_s be an optimal solution of objective function value Z^t_s.

Step 2.2: Solve problem $\mathrm{RS_p}(\lambda^t_s)$.

If the problem is finite, let x^t_s be a primal optimal solution, let $(\pi, m)^t_s$ be a dual optimal solution, and let $z(\lambda^t_s)$ be the objective function value of sub-problem. If $Z(\lambda^t_s) \leq Z^t_s$, then (x^t_s, λ^t_s) is an optimal solution to the master problem with constraints set Ω^1, and go to step3; otherwise, set $P^t_{s+1,D} := P^t_{sD} \cup \{(\pi, m)^t_s\}$, $R^t_{s+1,D} := R^t_{sD}$, $s := s + 1$, and return to step 2.1.

If the sub-problem is infeasible, let $(\pi, m)^s$ be a dual extreme ray such that

$$\sum_{t\in\{0,\ldots,T\}} \sum_{(i,j)\in I_1(A)} u_{ij}(t)m_{ij}(t)\lambda_{ij}(t) \le - \sum_{t\in\{0,\ldots,T\}} \left(\sum_{i\in N} \pi_i(t)b_i(t) + \sum_{(i,j)\in I_2(A)} u_{ij}(t)m_{ij}(t) \right)$$

Set $R^t_{s+1,D} := R^t_{SD} \cup \{(\pi, m)^t_s\}$, $P^t_{s+1,D} := P^t_{sD}$, $s := s + 1$, and return to step 2.1.

Step 3: If (x^t_s, λ^t_s) satisfies constraints (16), (x^t_s, λ^t_s) is the optimal solution to the original problem (P), **stop**; Let N^t be the set of nodes where shared information constraints (16) are violated. Set $\Omega^{t+1} := \Omega^t \cup N^t$, $t := t + 1$, and go to step 2.

The BD algorithm terminates with the optimal solution (Z_p) to problem (P). Step 2 ensures that (x^t_s, λ^t_s) is a feasible solution to problem (P), such that $Z(\lambda^t_s) \ge Z_p$ will hold. (λ^t_s, Z^t_s) is an optimal solution to the relaxation of problem $(\bar{P})$. Hence, $Z^t_s \le Z_p$ and if $Z(\lambda^t_s) \le Z^t_s$, then $Z(\lambda^t_s) = Z^t_s = Z_p$. Thus, as long as problem (P) is feasible, the algorithm will always terminate with an optimal solution (x^t_s, λ^t_s). It is well known that such Benders decomposition algorithms have exponential worst-case computational complexity, because it is possible that in the worst-case all the extreme points and extreme rays of $\boldsymbol{D}$ will be enumerated.

3.4. Example to Illustrate Nature of Solution

The solution of a small problem instance is shown to illustrate the nature of solutions to the BEPSI and to distinguish such solutions from typical solutions of other related network flow problems.

Specifically, solution to the BEPSI by the BD algorithm presented in Section 3.3 is compared with solution to the TDQFP by the extension of the TDQFP algorithm for multiple sources on a small time-dependent network given in Fig. 2.

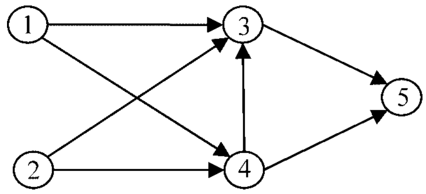


Figure 2. Example network.

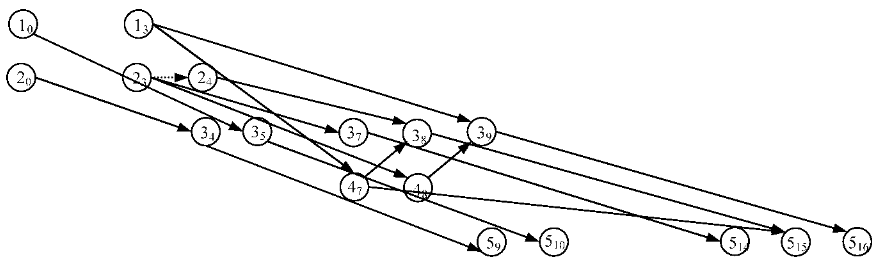
Assume that $T = 20, b_1(0) = 10, b_2(0) = 15, b_1(3) = 20, b_2(3) = 25, b_5(T) = -70$, and $b_i(t) = 0$, otherwise. A holdover arc, (i, i), exists at each $i \in N$. The time-dependent link traversal times and capacities are given in Table 1. Recall that for all $t \in \{0, \ldots, T\}$ and $i \in N\backslash l$, $\tau_{ii}(t) = 1$, and $u_{ii}(t) = \infty$.

The resulting solution to the BEPSI and related TDQFP are illustrated in Fig. 3 on a time-expanded network. The time-expanded network is created by making copies of the original network for each discrete interval of time. The numbers correspond to physical node numbers and their subscripts represent the departure time intervals, e.g. 2_4 represents node 2 at time 4. Waiting arcs are shown as dashed lines and are defined at every node between every consecutive pair of departure times.

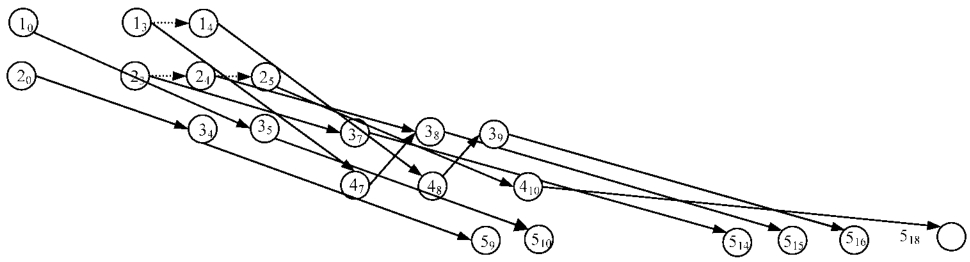
The example illustrates that the TDQFP solution may be infeasible for the BEPSI and that solution of the BEPSI is not necessarily optimal for an objective of minimizing the time by which the last evacuee exits the building. The TDQFP solution contains three nodes (nodes 1_3, 2_3, and 4_7) at which flow is split and is, therefore, an infeasible solution to the BEPSI. In the solution to the BEPSI (Fig. 3b), the last unit of flow exits the network at time 18. Since a solution exists for which it is possible that, for a greater total time, the time by which the last unit of flow exits the network can be reduced (i.e. from 18 to 17 units of time via 4_9 from node 2_4), it can be shown that triple optimization results given in [14] for a set of dynamic flow problems do not hold for the BEPSI. Specifically, optimal solution of the BEPSI is not necessarily optimal for an equivalent problem that seeks the minimum time by which the last unit exits the network in place of minimizing total time.

Table 1. Time-dependent travel times and capacities of example in Fig. 2.

(i, j)	(1, 3)	(1, 4)	(2, 3)	(2, 4)	(3, 5)	(4, 3)	(4, 5)
$\tau_{ij}(t)$	$5, t = 0$ $6, 1 \le t \le 20$	$4, 0 \le t \le 5$ $6, 6 \le t \le 20$	$4, 0 \le t \le 4$ $6, 5 \le t \le 20$	$4, 0 \le t \le 1$ $5, 2 \le t \le 10$ $7, 11 \le t \le 20$	$5, 0 \le t \le 6$ $7, 7 \le t \le 19$ $9, t = 20$	$1, 0 \le t \le 14$ $3, 15 \le t \le 20$	$6, 0 \le t \le 3$ $8, 4 \le t \le 20$
$u_{ij}(t)$	$20, 0 \le t \le 2$ $15, 3 \le t \le 20$	$20, 0 \le t \le 1$ $15, 2 \le t \le 6$ $10, 7 \le t \le 20$	$20, 0 \le t \le 1$ $10, 2 \le t \le 20$	$20, 0 \le t \le 2$ $15, 3 \le t \le 20$	$25, 0 \le t \le 2$ $20, 3 \le t \le 20$	$20, 0 \le t \le 9$ $18, 10 \le t \le 17$ $15, 18 \le t \le 20$	$25, 0 \le t \le 12$ $20, 13 \le t \le 20$



(a) Solution to the multisource TDQFP



(b) Solution to the BEPSI

Figure 3. Final solutions to the time-dependent evacuation problem with and without shared information constraints.

PROPOSITION 2: The value of the optimal solution to the multi-source TDQFP provides a lower bound on the value of the optimal solution to the BEPSI.

DISCUSSION: The feasible region of the BEPSI is contained in the feasible region of the multisource TDQFP and, is thus, more restrictive than that of the multi-source TDQFP. Hence, the value of the optimal solution to the multi-source TDQFP provides a lower bound on the value of the optimal solution to the BEPSI.

4. COMPUTATIONAL EXPERIMENTS

Results of computational experiments conducted on a network representation of an existing, four-story building, the A. V. Williams Building, on the University of Maryland campus are given in this section. Data for the building was collected on-site, taking actual measurements of doorways, corridor widths and lengths, stairwell widths, and other dimensions. The layout of the four floors was similar; thus, data was only collected on the second floor and was replicated to create the network model of the four-story building. The layout of the second floor is shown in Fig. 4.

4.1. Experimental Design

A network representation of the A. V. Williams Building was developed by placing nodes on each side of each doorway connected by an edge to allow the movement of people between rooms and corridors, into and out of stairwells and through the exits and by placing nodes at the intersection of corridors. The nodes in the corridors were connected by edges. Edges were also used to represent stairwells. Elevators were ignored, because use of elevators in this building is prohibited during an evacuation. It was further assumed that escape from the first floor was only possible through doorways; no window egress was modeled. This resulted in a 612-node, 1,480-edge network with five exit nodes. The maximum occupancy related to the classrooms, offices, laboratories, and lavatories permitted by fire codes were estimated

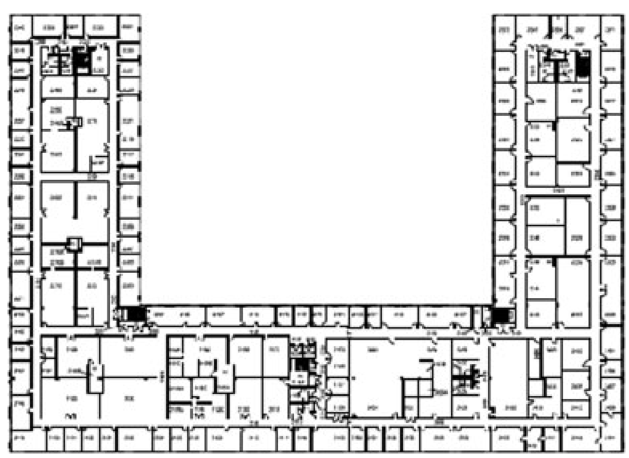


Figure 4. The A. V. Williams building second floor layout.

Table 2. Crowd movement parameters for various facilities.[a]

Facility	Density (person/ft^2)	Speed (ft/min)	Flow (person/min/ft)
Doorway	0.22	120	26
Pathway	0.20	120	24
Stairwell	0.19	95	18

[a]Ref. [7]

with the use of the 2000 edition of the NFPA 101 Life Safety Code [26].

The amount of supply (i.e. evacuees) at each node is set based on variations of the maximum occupancies of the rooms in the building as per the NFPA Life Safety Code. Three levels of supply are considered (average, maximum and maximum plus), where the maximum plus category introduces exceptional supply levels at a subset of critical nodes.

Two approaches were considered for estimating flow rates that can be translated to travel times and capacities associated with the edges. The first is to calculate the saturated flow rate from empirical formulae that have been proposed in the literature (see, for example, [4]). The second is to use values related to pedestrian movement characteristics provided in the SFPE Handbook of Fire Protection Engineering (1988). The latter approach was employed in estimating these values for the A. V. Williams Building. These estimates are provided in Table 2.

Edge capacities were set to the maximum flow rate as computed from rates given in Table 2. The time interval duration for time discretization was assumed to be 1 min. Speeds were employed to estimate edge travel times.

Six scenarios were considered in tests of the BD algorithm for solving the BEPSI that were conducted on the network representation of the A. V. Williams Building. The factors that were considered in the construction of these scenarios include the number of people present at the time of the initiation of the evacuation (i.e. supply at the source nodes), whether or not corridors and stairwells were blocked or impaired (i.e. whether or not edges were operating at maximum capacity and maximum speeds could be reached), and the type and location of the event triggering the evacuation.

In the first two scenarios, conditions were assumed to be ideal, as would be the case in a fire drill as opposed to an actual fire. Conditions were, therefore, assumed to be time-invariant. Multiplication factors were applied to this ideal scenario in the remaining three scenarios to replicate conditions that were worsening over time. The multiplication factors are cumulatively applied from one time interval to the next to both capacities and travel times and are given in Table 3.

Scenarios are designed such that the scale of the hazard that initiated the need for the evacuation and its impact increase with increasing scenario identification number. In scenarios 3 to 5, conditions are assumed to be worse than those of the ideal scenarios (scenario 1 and 2), but no specific hazard location is simulated. However, in scenario 6, the hazard is assumed to occur at a location that results in untenable conditions or blockages along major escape pathways. In this scenario, it is assumed that a fire begins in the west wing of the fourth floor. Conditions deteriorate rapidly. One corridor in the west wing is blocked and the nearest stairwell is impassable.

In all six scenarios, time horizon T was assumed to be 20 min and stairwells and corridors were assumed to be empty at initiation of the evacuation. Results from application of the BD algorithm on the A. V. Williams Building under these six scenarios are discussed next.

4.2. Result Analysis

The BD algorithm was implemented in Microsoft Visual Studio C++ 6.0 language with the ILOG CPLEX callable library 9.1 [13] and was run on a personal computer with Pentium (4) CPU 3.20 GHz and 2.00 GB of RAM.

Valid cuts (18) are added to the Benders master problem ($\bar{\mathrm{P}}$) to accelerate convergence to the optimal solution. At each step, where an optimality cut is desired, a Pareto-optimal cut is generated. It was observed in the experiments that these cuts led to quick convergence on the optimal solution. For most of the problem instances that were tested, the number of iterations and computation time were reduced considerably by the inclusion of the Pareto-optimal cuts as compared with runs in which these cuts were not employed. Additional computational improvements might be obtained by relaxing integrality constraints on the variables of the

Table 3. Characteristics of test scenarios.

Scenario	Capacities	Travel times	Supply level	Severity of conditions
1	1	1	1	Ideal conditions
2	1	1	3	Ideal conditions
3	0.98	1.02	1	Slightly impacted
4	0.98	1.02	2	Slightly impacted
5	0.96	1.04	3	Impacted
6	0.95	1.06	3	Severely impacted, some links disabled

Table 4. Computational results for the real-world network.

			Computational time (CPU seconds)		
			BD		
Scenario	$\Delta(Z_{\text{BEPSI}} - Z_{\text{TDQFP}})$	Number of cuts	To 95% optimality	To optimality	Branch-and-cut
1	0	4	—	3.0	4.6
2	0	4	1.6	3.3	21.7
3	0	12	1.9	30.8	80.0
4	32	36	6.0	31.2	178.7
5	0	32	19.6	58.5	221.3
6	224	44	17.7	94.8	> 0.5 h

relaxed master problem and generating Benders cuts from fractional solutions as was proposed by McDaniel and Devine (18). McDaniel and Devine showed that exact solution of the relaxed master problem was not required at each step and noted that any feasible solution can generate Benders cuts.

The results of experiments showed that there is a significant reduction in computational time obtained by using sub-problem $(\text{RS}_\text{p}(\tilde{\lambda}))$ instead of $(\text{S}_\text{p}(\tilde{\lambda}))$. CPU times were reduced by a factor of at least 10 for all tested cases. Either a generic MIP solver or specially designed algorithms, such as the TDQFP algorithm, can be employed to solve $(\text{RS}_\text{p}(\tilde{\lambda}))$. CPLEX's MIP solver is applied in this work.

An alternative to the BD algorithm is to employ a branch-and-cut algorithm based on a similar concept to the relaxation step employed in the BD algorithm (i.e. step 3). As illustrated in the example in Section 3.4, solution of the TDQFP may result in split flows at one or more locations. Let the set of the nodes where flow splits in the TDQFP solution be $S^0(N) \subset N$. A set of constraints can be generated to enforce unsplittable flow as follows.

$$\sum_{j \in \Gamma^+(i), j \neq i} x_{ij}/u_{ij} \leq 1, \quad \forall i \in S^0(A). \tag{21}$$

If the current solution violates cuts (21), then the cuts are valid. Repeat the process until no valid cut can be generated. Once a solution is obtained that does not violate cuts (21), branch on the x_{ij}, $(i,j) \in A$, variables that violate the shared information constraints (4), i.e. impose the disjunction $(x_{ij^1} = 0) \vee (x_{ij^2} = 0) \vee \ldots \vee (x_{ij^s} = 0)$, where $x_{ij^1}, x_{ij^2}, \ldots, x_{ij^s} \in \{x_{ij} | x_{ij} > 0, \forall i \in S^0(A)\}$. The number of branches s equals the number of arcs with positive flow departing from the same node at the same time.

The computational time required by the BD algorithm, as well as the branch-and-cut technique, for solving the BEPSI in the A.V. Williams Building is provided in Table 4. The scenario number as defined in Table 3 is given in the first column. The second column reports the difference between the optimal objective function value to the BEPSI, containing shared information constraints, and the TDQFP (extended for multiple origins), where the shared information constraints are dropped. The third column reports the number of iterations, i.e. number of Benders cuts. The fourth and fifth columns report the computational time in CPU seconds used by the BD algorithm to reach 95% of optimality and optimality, respectively. The computational time required by the branch-and-cut algorithm to reach optimality is given in the sixth column. All reported times include all input and output time.

Results show that as the problem becomes more difficult and waiting arcs are required, the required computational time to solve the problem to optimality by either approach increases. The more frequent flow splits in the TDQFP solution, the greater the computational effort required by the BD and branch-and-cut algorithms. It is also postulated that the performance of both algorithms will be impacted by the degree of each node, because the larger the degree, the more likely flow is to split. The required computational time of the BD algorithm increases less than linearly with increasing supply and deteriorating network conditions. Moreover, the computational time required to achieve 95% of optimality is significantly less than that required to achieve optimality. Since the BD algorithm can be prematurely terminated with a feasible solution, stopping the algorithm after a short period of time may be a viable alternative. In all scenarios, the BD algorithm outperforms the branch-and-cut method. Note that step 3 of the BD algorithm is specialized for this particular application. It was observed that the addition of step 3 to the BD algorithm, where only a subset of constraints (16) of problem $(\bar{\text{P}})$ are enforced, led to significant reductions in computation time. Additional experiments would be required to assess the impact of network size on the computational performance of these techniques.

In building evacuation, as new information about the current state of the building's structures and circulation systems are obtained, updates to the network model in terms of supply, arc capacities and arc traversal times will be made and a new BEPSI will need to be solved. Rather than starting from scratch, it is possible to employ the Benders cuts generated in the prior problem instance as the initial cuts in employing the

Table 5. Reoptimization results of the BD algorithm.

	Increase of supply		Decrease of capacity	
	Select nodes	Entire network	Select arcs	Entire network
Computational time required with reoptimization (CPU seconds)	33.4	33.5	30.5	24.6
% of time required as compared to resolving from scratch	41.5%	42.0%	37.9%	30.5%

BD algorithm to solve the new problem instance if the supply increases and/or arc capacities decrease. Decreases in arc capacities are expected in circumstances warranting an evacuation, as fire and smoke will spread throughout the building and collapse of the structural components will occur progressively. That is, conditions worsen with time and capacities accordingly decrease with time. Additional experiments were conducted to assess the magnitude of improvement that results from employing the Benders cuts generated in the prior problem instance in solving the updated problem.

Changes to arc capacities and supply in Scenario 3 were considered in these additional experiments. Specifically, four updates were considered: (1) supplies at randomly chosen nodes increase, (2) supply at all supply nodes increase, (3) capacities of randomly chosen arcs decrease, and (4) capacities of all arcs decrease. One might also assess the benefits of such a reoptimization approach where changes in supply and capacities occur simultaneously. Results of runs on these variants are given in Table 5.

The results show that significant (on the order of 60–70%) reductions in computational time result from solving the updated problem instance starting with the Benders cuts generated in solving the prior problem instance (i.e. the reoptimization time) when compared with solving the new problem instance from scratch (i.e. with no information from the prior problem instance).

5. CONCLUSIONS AND FUTURE RESEARCH

In this article, the BEPSI is formulated as a mixed integer linear program. The problem is shown to be NP-hard. An exact algorithm based on Benders decomposition is proposed for its solution. Computational experiments performed on a network representation of an actual four-story building were conducted to illustrate how the proposed procedure can be applied to solve for the optimal evacuation instructions in an actual building and to demonstrate the feasibility of its application. The solution technique is designed in such a way that it can be prematurely terminated and feasible solutions can be obtained. Experimental results show that significantly less time is required to obtain solutions that are within 95% of optimality.

By restricting flows to a single arc at each point in time and explicitly considering the inherent dynamic nature of future conditions, the resulting evacuation plans are more likely to be followed in light of our understanding of group dynamics in evacuation and to aid the evacuees in avoiding potentially high risk situations. Traditional evacuation planning techniques ignore the dynamics of a fire moving through a corridor or through a stairwell and existing optimization techniques would not prevent solutions from suggesting groups to split at the nodes. Consequently, implementation of evacuation plans developed by the proposed technique for a large building, ship, or military complex can result in a reduction in the number of lost lives, trapped evacuees or rescue workers, and risk of exposure. Further, shorter egress times may result, permitting recovery efforts to begin quickly.

As presented, solution of the proposed formulation may result in flows that arrive at an intermediate location at a given point in time, but depart along different paths by departing at different departure time intervals, i.e. by definition, the flow is not split, but in practice, the flows take different paths. This type of splitting of flows is permitted through the introduction of holdover arcs that are modeled to ensure feasibility. If such holdover arcs were not permitted, it would be difficult to model situations where there is an excess of evacuees waiting to enter a chosen path with insufficient capacity to handle all evacuees who arrive in a single time interval. In an evacuation, conditions typically worsen with time; that is, the arc traversal times are FIFO. Thus, it is always best to leave as early as possible and waiting will not be chosen if it can be prevented. Additionally, capacity of the holdover arcs may be restricted and the discretization interval size can be set to a sufficiently large value to minimize the occurrence of such splitting of flows.

One might argue that arc traversal times are not only a function of the network conditions, but are a function of the number of people simultaneously using the arcs. Such load-dependent flows are considered in [17] with respect to the quickest flow problem. This concept of selecting paths such that flows are not split can be extended to consider flow-dependent traversal times.

The procedures developed through the proposed research activity will impact many other functional areas as well, including, for example, evacuation of a geographic region due to military attack, human-made accident, or natural disaster, such as an accident involving a nuclear power plant or escape of hazardous chemicals, collapse of a structure such as dam walls, hurricane, earthquake, flooding, volcanic eruption, or tsunami. Evacuation instructions can be provided to vehicles via changeable message signs, radio, the internet, or on-board devices in suitably equipped vehicles with further development of Intelligent Transportation Systems. Moreover, as with other network flow-based techniques, it is expected that the techniques proposed herein will have application in many diverse arenas, such as production-distribution systems, fleet management, and communications.

Many theoretical and practical aspects of this problem remain to be explored. For some problem instances, or building layouts, it may be feasible to employ the TDQFP algorithm or similar that allows splitting of flow, if the solutions are unlikely to contain split flows. Heuristic repair operators can be applied to locations of split flow to obtain feasible and potentially near-optimal solutions. Experiments on additional building designs could be conducted to assess the negative impact on total evacuation time that results from enforcing solutions that do not permit splittable flows. Finally, heuristics could be developed to more quickly obtain feasible and, hopefully, near-optimal solutions for large-size networks. The exact procedure proposed herein for this difficult problem can be used to obtain benchmark solutions, enabling evaluation of quicker, heuristic techniques.

Evacuees may not prefer the solution that provides the minimum evacuation time or that optimizes other functions of time, but instead may prefer a path with high likelihood of leading to successful escape. Alternative objectives that consider these and other issues of equity that arise in solutions for the evacuation problem have been proposed in the literature (e.g. [19,27]). Regardless of the objective that is chosen for the determination of the optimal instructions, the issue of shared information arises if instructions are to be provided by changeable message signs or comparable technologies that inform more than one person at a time. One may extend this work to address the issue of unsplittable flows in the context of other objectives, such as those related to minimization of risk or time by which the last evacuee safely exits.

ACKNOWLEDGMENTS

This work was supported by NSF grant CMS 0348552. This support is gratefully acknowledged, but implies no endorsement of the findings.

REFERENCES

[1] N. Baumann and M. Skutella, Solving evacuation problems efficiently–earliest arrival flows with multiple sources, Proc 47th IEEE Symp Foundations of Comput Sci 2006, pp. 399–410.
[2] J.F. Benders, Partitioning procedures for sloving mixed-variables programming problems, Numer Math 4 (1962), 238–252.
[3] A. Chakrabarti, C. Chekuri, A. Gupta, and A. Kumar, Approximation algorithms for the unsplittable flow problem, Lecture Notes Comput Sci 2462 (2002), 51–66.
[4] L.G. Chalmet, R.L. Francis, and P.B. Saunders, Network models for building evacuation, Management Sci 28 (1982), 86–105.
[5] J.F. Cordeau, F. Soumis, and J. Desrosiers, A Benders decomposition approach for the locomotive and car assignment problem, Transport Sci 34 (2000), 133–149.
[6] A.M. Costa, A survey on benders decomposition applied to fixed-charge network design problems, Comput Oper Res 32 (2005), 1429–1450.
[7] P.J. DiNenno, R.L.P. Custer, W. Walton, and J.M. Watts, The SFPE handbook of fire protection engineering, 1st ed., The National Fire Protection Association, 1988.
[8] Y. Dinitz, N. Garg, and M.X. Goemans, On the single-source unsplittable flow problem, Combinatorica 19 (1999), 17–41.
[9] B. Fleischmann, M. Gietz, and S. Gnutzmann, Time-varying travel times in vehicle routing, Transport Sci 38 (2004), 160–173.
[10] M.R. Garey and D.S. Johnson, Computers and intractability: A guide to the theory of NP-Completeness, W. H. Freeman, San Francisco, CA, 1979.
[11] H.W. Hamacher and S.A. Tjandra, "Mathematical modelling of evacuation problems: A state of art," Pedestrian evacuation dynamics (2002), Springer, Berlin Heidelberg, 227–266.
[12] D. Helbing, I. Farkas, and T. Vicsek, Simulating dynamical features of escape panic, Nature 407 (2000), 487–490.
[13] ILOG CPLEX Callable Library 9.1 reference manual (April 2005).
[14] J. Jarvis and H. Ratliff, Some equivalent objectives for dynamic network flow problems, Management Sci 28 (1982), 106–109.
[15] N.R. Johnson, Collective behavior as group-induced shift, Sociol Inquiry 44 (1974), 105–110.
[16] N. Kamiyama, N. Katoh, and A. Takizawa, An efficient algorithm for evacuation problem in dynamic network flows with uniform arc capacity, IEICE Trans Inform Syst E89-D (2006), 2372–2379.
[17] M. Köhler and E. Skutella, Flows over time with load-dependent transit times, SIAM J Opt 15 (2005), 1185–1202.
[18] S.G. Kolliopoulos and C. Stein, Approximating disjoint-path problems using packing integer programs, Math Program Ser A 99 (2004), 63–87.
[19] Y.K. Lin, A simple algorithm for reliability evaluation of a stochastic-flow network with node failure, Comput Oper Res 28 (2001), 1277–1285.
[20] G.G. Lovas, Models of wayfinding in emergency evacuation, Eur J Oper Res 105 (1998), 371–389.
[21] Q. Lu, Y. Huang, and S. Shekhar, Evacuation planning: A capacity constrained routing approach, Comput Sci 2665 (2003), 111–125.
[22] T.L. Magnanti and R.T. Wong, Accelerating Benders decomposition: Algorithmic enhancement and model selection criteria, Oper Res 29 (1981), 464–484.
[23] S. Mamada, K. Makino, T. Takabatake, and S. Fujishige, The evacuation problem, dynamic network flows, and algorithms, SICE Annual Conference in Fukui, 2003, pp. 2807–2811.
[24] E. Miller-Hooks and T. Krauthammer, An intelligent evacuation rescue and recovery concept, Fire Technol 43 (2007), 107–122.

[25] E. Miller-Hooks and S. Stock Patterson, On solving quickest time problems in time-dependent, dynamic networks, J Math Model Algorithms 3 (2004), 39–71.
[26] National Fire Protection Association, Inc..1 Battery- march Park (2001).
[27] S. Opasanon and E. Miller-Hooks, The safest escape problem, in press, J Oper Res Soc (2007).
[28] J.D. Sime, The outcome of escape behavior in the Summerland fire: Panic or affiliation?, International Conference on Building Use and Safety Technology, Los Angeles, CA, 1985.
[29] S.A. Tjandra, Dynamic network optimization with application to the evacuation problem, Fachbereich Mathematik der Universität Kaiserslautern, 2003.
[30] D. Wenger, B. Aguirre, and G. Vigo, Evacuation behavior among tenants of the World Trade Center following the bombing of February 26, Hazard Reduction Recovery Center Publications, Texas A& M University, 1994.
[31] D. Wenger and G. Vigo, A test of emergent norm theory of collective behavior, Sociol Forum 13 (1998), 301–320.