IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received September 23, 2019, accepted November 12, 2019, date of publication November 25, 2019,
date of current version December 6, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2955477

An Efficient Building Evacuation Algorithm in Congested Networks

CHANG HYUP OH[1], MIN HEE KIM[2], BYUNG-IN KIM[1], AND YOUNG MYOUNG KO[1]
[1]Department of Industrial and Management Engineering, Pohang University of Science and Technology, Pohang 37673, South Korea
[2]Department of Industrial and Systems Engineering, University of Wisconsin–Madison, Madison, WI 53706, USA

Corresponding author: Young Myoung Ko (youngko@postech.ac.kr)

This work was supported in part by the Korea Institute for Advancement of Technology (KIAT) Grant funded by the Korean Government (MOTIE) (The Competency Development Program for Industry Specialist) under Grant N0008691.

ABSTRACT This paper proposes a new network model for the building evacuation problem considering congestion levels and provides a mixed integer linear programming (MILP) model and an efficient heuristic algorithm solving the problem. Constructing an optimization model with several congestion levels, we introduce a new network called the multi-class time-expanded (MCTE) network having several exclusive arcs connecting the same tail and head nodes. The MCTE networks make both the MILP model and the heuristic algorithm reflect a realistic situation in congested networks. Considering MCTE networks makes the problem difficult to solve, which motivates us to develop an efficient heuristic algorithm. We test our heuristic algorithm using several real-world networks such as a multiplex cinema, a subway station, and a large-size complex shopping mall in addition to an artificial network for clear comparison between the proposed algorithm and the MILP approaches. The results indicate that the proposed algorithm runs fast and produces a near-optimal solution compared with those from MILP models with a commercial solver.

INDEX TERMS Building evacuation, congested networks, movement evacuation model, multi-class time-expanded networks.

I. INTRODUCTION

Recent tragedies such as the Manchester Arena terrorist attack, the Grenfell Tower fire in London and the Bataclan concert hall attack in Paris are motivating the development of time-critical evacuation plans. The fact that in less than 10 minutes fire engulfed the 24-story Grenfell Tower underscores the need for an efficient evacuation algorithm that can run fast enough to account for the rapidly changing status of places to evacuate in emergencies. The Internet of Things (IoT) technology enables us to collect and share essential building information such as the number of evacuees in each space and availability of each space in real-time and plays as an essential infrastructure for implementing real-time evacuation plans from data.

Inspired by such technological advancement, we set our goal to develop an algorithm adequate for emergent evacuating situations. In developing such an algorithm, we notice that we should seriously consider congestion that can happen while people are evacuating. It is not hard to imagine that extreme congestion is likely to occur when evacuees take the same evacuation path. In addition to congestion itself, we also pay attention to the level of congestion that affects travel times even if the arc capacity is not fully occupied. For example, if a small number of evacuees use an arc, they can move fast through the arc. If many evacuees use the channel at the same time, however, their speed can be slower. To bring congestion levels into our problem, we propose a new network model, which we call a multi-class time-expanded (MCTE) network. An MCTE network, which will be explained in Section III-A, consists of multiple arcs corresponding to different congestion levels (multi-class property) and copies of nodes over time considering multiple evacuation periods (time-expanded property). Each arc has different travel time and capacity that correspond to the congestion level between two nodes. Fig. 1 is a simple schematic example having three different classes of arcs between nodes i and j.

A. CONTRIBUTIONS AND ORGANIZATION OF THE PAPER

This paper makes the following contributions to the evacuation planning literature. First, we propose a new network model (the MCTE network) and build a mixed integer linear

The associate editor coordinating the review of this manuscript and approving it for publication was Rajesh Kumar.

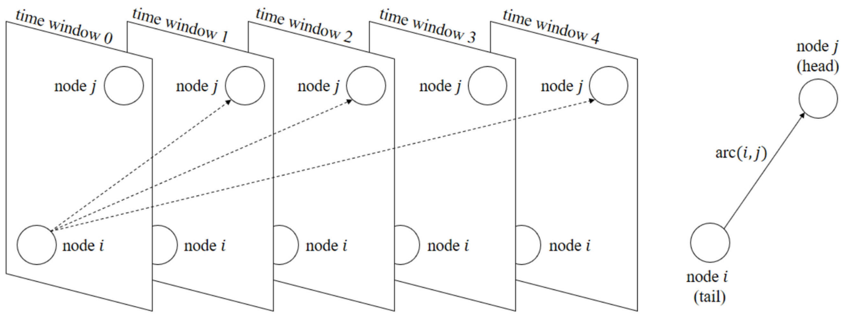


FIGURE 1. **Nodes and arcs in an MCTE network.**

programming (MILP) model incorporating congestion levels. In the optimization model, constraints related to congestion levels are nonlinear, and we convert them into linear ones so that we can take advantage of the computing power of commercially available MILP solvers.

Second, for scalability, we develop a heuristic algorithm providing near-optimal solutions for both small and large-size problems. We note that the proposed algorithm can apply to both planning and operational problems. The problem in a MILP form is usually designed for planning. One, however, can solve operational problems by iteratively solving the problem, for example, using moving windows. For such cases, coming up with quick solutions should be critical. Since fire and smoke spread out every second in a fire situation, an evacuation algorithm should solve problems and deliver solutions to instruction devices in a few seconds.

We emphasize that the preliminary version of the proposed algorithm has been running in real-world networks. The subway station and the shopping mall in Section VI-C and VI-D are using it with the IoT-based instruction devices.

From extensive numerical studies, we show that the heuristic algorithm is significantly faster than the MILP models with solvers. Third, we apply our heuristic algorithm not only in an artificial network but also in real-world building networks including cinema, subway station, and shopping mall networks for better implementation in realistic situations.

The remainder of this paper is organized as follows. Section II reviews the evacuation planning literature. Section III gives the problem details and introduces the MCTE network model. Section IV constructs the MILP model for the MCTE network. Section V explains our heuristic algorithm. Section VI describes the dataset, the simulated and real building networks, and discusses the numerical results. Section VII concludes and suggests future research.

II. LITERATURE REVIEW

The fundamental framework introduced by [1], known as building 101, considered the capacity of nodes and arcs, the travel times of arcs, and the supplies of nodes. The model determined an evacuation route for evacuees that minimized evacuation time, based on the time-expanded network model suggested by [2] for congested situations.

Choi *et al.* [3] suggested a network flow formulation with flow-dependent arc capacity constraints by defining the capacity of each arc as a function of the number of evacuees in its tail node. Hamacher and Tufekci [4] showed how to avoid unnecessary movement like cyclic movements in a building. As explained in [3] and [4], a time-expanded network allowed pseudo-polynomial algorithms. Hoppe and Tardos [5] proposed a polynomial time algorithm for dynamic network flow problems in the time-expanded network with multiple sources and sinks.

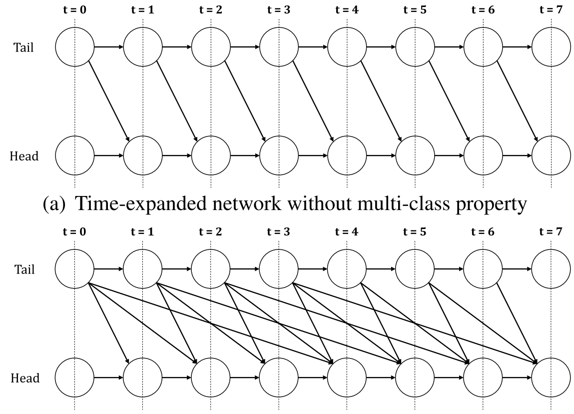
Unfortunately, the size of the time-expanded networks imposes a computational burden on the studies cited above. Thus, several heuristic algorithms have been developed, i.e., the Single-Route Capacity Constrained Planner (SRCCP) and the Multi-Route Capacity Constrained Planner (MRCCP) algorithms proposed by [6]. Lu *et al.* [7] later developed the enhanced algorithm known as the Capacity Constrained Route Planner (CCRP) whose running time was $O(p \cdot n \cdot \log n)$. Mishra *et al.* [8] proposed a Source Single Sink Evacuation Route Planner, whose final evacuation time is always better and whose computation time is shorter than CCRP.

The CCRP algorithm uses a generalized Dijkstra's shortest path algorithm that needs constant capacities of nodes and arcs to find evacuation paths. While assuming constant capacities makes an algorithm faster, it does not consider congestion levels that may significantly affect evacuation time through arcs. Kim *et al.* [9] proposed the Intelligent Load Reduction (ILR) and the Incremental Data Structure (IDS) algorithms. The ILR algorithm reduces the load by diverting routes to destinations when a larger number of evacuees is in a bottleneck, and the IDS algorithm reduces the calculation time by using the improved data structures without changing the outputs of the CCRP algorithm. Cepolina [10] used the concept of the dynamic capacities of arcs introduced in [3].

Lin *et al.* [11] proposed the multi-state time-varying quickest flow (MSTVQF) algorithm for the time-varying multi-source-multi-sink quickest network flow problem. MSTVQF includes important characteristics of realistic situations such as multi-source-multi-sink and time-expanded properties. Lin *et at.* [11] also suggested the concept of phased evacuation, which assigns an evacuation order to all nodes depending on higher and normal order priority. Koo [12] and

Noh *et al.* [13] used the phased evacuation concept to solve the evacuation problem with a heterogeneous population of normal and disabled evacuees. The strategy can mitigate or reduce congestion caused by disabled evacuees.

In this paper, we define the capacities and travel times of arcs according to the number of evacuees on the arcs. It is natural to think that the more evacuees are on an arc, the slower the evacuees travel because of congestion. Fruin [14], [15] investigated the relationship between the number of people in a unit area and the walk speed. Fruin [14] provided graphs illustrating the relationship between population density and walking speed in various situations. We use their results to create multi-class arcs explaining the relationship between the number of evacuees and the travel time between two nodes.



(a) Time-expanded network without multi-class property

(b) Time-expanded network with multi-class property

FIGURE 2. Time-expanded network with/without multi-class property.

III. PROBLEM DESCRIPTION

Our aim is to minimize the final evacuation time. We define the final evacuation time, also known as the network clearance time, as the time when all evacuees have successfully moved to safety zones or beyond the building. The problem may have several objectives such as minimizing final evacuation time, minimizing the average evacuation time, or maximizing the number of people who move to exits within a given time limit. Jarvis and Ratliff [16] showed that it is possible to achieve multiple objectives simultaneously, and the objectives are mathematically equivalent. In the following sections, we propose the MCTE network and describe input data for generating an MCTE network from building information.

A. NETWORK

We consider a time-expanded network, as described in [4]. Each node has its capacity and the number of initial evacuees, i.e., node supplies. The capacity of a node indicates the maximum number of evacuees who can stay at the node in each time window. We call nodes having at least one evacuee at time 0 as source nodes. All nodes except exit nodes can be source nodes if there are evacuees at the beginning. Each arc requires information about its capacity, head/tail nodes, and travel time. The arcs are the passageways connecting head and tail nodes. The capacity of each arc is the maximum number of evacuees held by each arc at the same time. The capacity relates to the width of the passageway such as an aisle, stairwell, hallway, etc. Travel time is the amount of time needed by an evacuee to move from an arc's tail node to head node. The length and the width of the passageway affect its travel time. We obtain capacities and travel times from the floor plans of a building.

We would now consider congestion levels: the relationship between travel time and the number of evacuees assigned to an arc. Time-expanded networks themselves, however, are not ready for this purpose, so we extend the time-expanded network so that each pair of two different spaces has multiple arcs (classes) representing congestion levels. Fig. 2 shows a time-expanded network without a multi-class property and our proposed MCTE network.

We consider two nodes in Fig. 2. The tail node at each time t in the time-expanded network has only one transit arc to the head node, whereas the MCTE network has multiple arcs with different capacities and travel times. In the MCTE network, we call the arc having the smallest capacity and travel time a *base arc* or a *class 1 arc*. The capacities and travel times of the other arcs are defined relative to the base arc. If the base arc has capacity μ and travel time λ, we construct the class k arc having capacity $k \cdot \mu$ and the travel time $w_k \cdot \lambda$, where w_k is obtained from the appropriate function considering the characteristics of the building, evacuees, etc.

According to [14], the relationship between pedestrians' mean speed (S) and mean density (M) is $S = (267M - 722)/M$. Since we want to find the relationship between arc travel time (T) and arc capacity (C), we change the equation in [14] into $T = (l \cdot V)/(267V - 722C)$, where l is the arc length and V is the arc volume.

To build an MILP model for the MCTE network, we discretize the function in [14] and obtain weights of arc classes (w_k). Note that we can use any other functions depending on the context of the problem. Fig. 3 shows a relationship between arc travel time and the number of evacuees on arc whose length is 20 meters and width is 3 meters. For the case in Fig. 3, if we assume the capacity of the base arc is 10, we can obtain the discretized approximation of $w_k = \{w_1, w_2, w_3\}$ to be 1, 2, and 4.

Fig. 2 shows simple schematic time-expanded and MCTE networks In Fig. 2(a), the tail node at time t connects to the head node at time $t+1$. In Fig. 2(b), every node has three transit arcs. The tail node at time 0 has three different arcs to the head nodes at times 1, 2, and 4. Each arc from the tail node at time t has a different travel time, which is a non-decreasing (nonlinear) function of its capacity. One thing we have to be careful in MCTE networks is that an undesirable situation that we call the FIFO (First-In-First-Out) violation may occur. A FIFO violation happens when some evacuees who leave the tail node later overtake other evacuees who leave the tail node earlier. Fig. 4 illustrates a case, and we will address it in Sections IV and V.

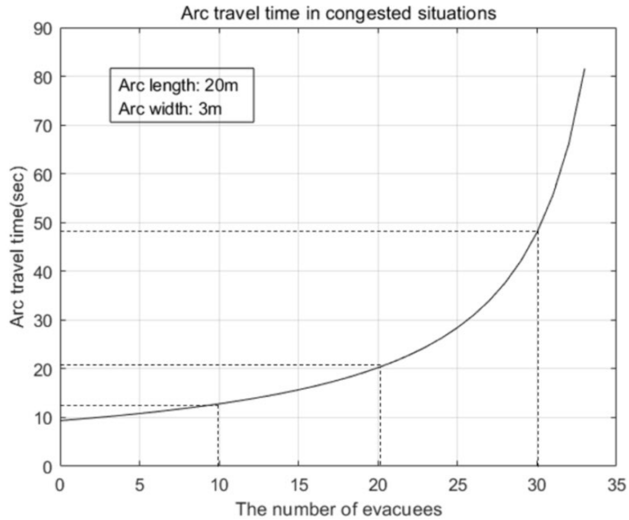


FIGURE 3. Relationship between arc travel time and the number of evacuees.

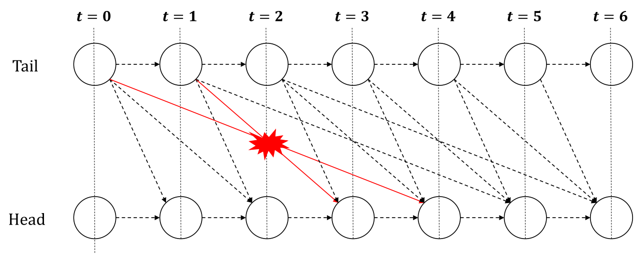


FIGURE 4. FIFO violation in MCTE networks.

TABLE 1. Node and arc information.

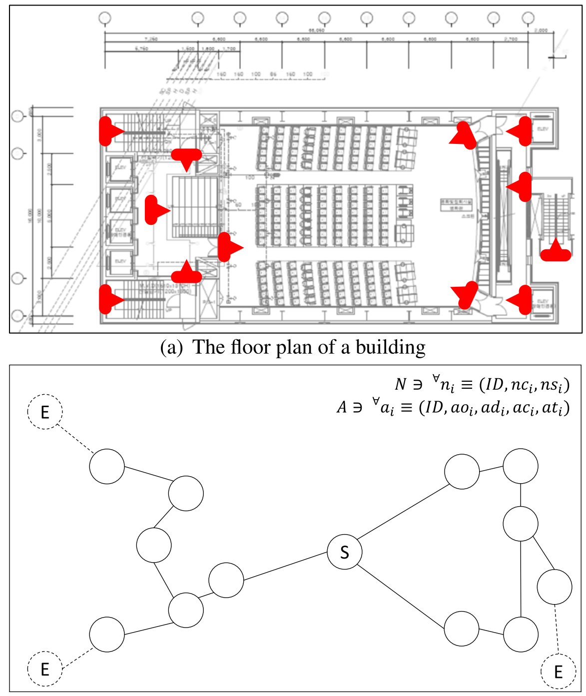
Node ID (Integer)	Floor (Integer)	Capacity (Integer)	Supply (Integer)	State (Binary)
1	1	15	10	True
2	2	10	0	True
3	2	10	0	False
...	...	...	...	...
N	5	10000	0	True

Arc ID (Integer)	Head (Integer)	Tail (Integer)	Capacity (Integer)	Travel Time (Integer)
1	1	2	5	7
2	2	1	5	7
...				
A	N-1	N	10	8

B. INPUT DATA AND OUTPUT DATA

Before explaining mathematical models, we briefly explain the data we use to construct them. We generate the MCTE network from the floor plan of a building, as shown in Fig. 5(a). The floor plan contains data on the dimensions of rooms, corridors, stairs, etc.

We use the dimensions and some additional information about the building in Table 1 to generate the skeletal networks in Fig. 5(b). The output includes final evacuation time, number of total evacuees, and evacuation paths.



(a) The floor plan of a building

(b) Network generation from the floor plan

FIGURE 5. The example of the generated network.

IV. MILP MODELS

This section describes the MILP models we build for finding evacuation paths in time-expanded networks and MCTE networks. We formulate an optimization problem which has a quadratic constraint for addressing the FIFO violation and then apply a linearization technique to change the problem to an MILP problem. Since the MILP problem for MCTE networks can be solved only for small-size networks, we add simpler MILP models that consist of only one class of arcs and use their solutions as upper bounds.

A. MILP MODEL FOR THE MCTE NETWORK

Lin *et al.* [11] suggested the MSTVQF algorithm for the multi-source multi-sink time-varying quickest network flow and phased evacuation planning problems. Koo [12] and Noh *et al.* [13] modified the formulation in [11] to find the optimal evacuation plan with their particular objectives. We build our formulation based on [11] and [13] and add the multi-class property for MCTE networks. We use the following notations:

N	set of all nodes in the network
S	set of all source nodes in the network
A	set of all arcs in the network
T	maximum evacuation time
$\delta^-(i)$	set of all precursor nodes of node i
$\delta^+(i)$	set of all successor nodes of node i
c_i	capacity of node i
$r(i)$	number of evacuees in node i at time 0
μ_{ij}	base arc capacity of the arc from node i to node j

λ_{ij} base travel time of the arc from node i to node j
w_k weight of class k arcs for travel time
d index of super-sink node

Decision variables

$x_{ijk}(t)$ number of evacuees who move from node i to node j using the class k arc at time t
$y_i(t)$ number of evacuees who stay at the node i in time interval $[t, t+1)$
$z_{ijk}(t)$ binary variable that indicates whether class k arc between node i and node j is used or not at time t

Jarvis and Ratliff [16] showed that the three objectives of the building evacuation problem (minimizing the network clearance time (T), maximizing the number of evacuees for the first p periods for each $p \leq T$, and minimizing the average time to evacuate the building) are equivalent. Although we choose the first one (the network clearance time), we minimize the sum of evacuees' evacuation times (equivalent to minimizing the average evacuation time) for the MCTE network, to avoid solving a min-max problem that minimizes the maximum evacuation times of evacuees. We formulate the MILP model as follows:

$$\underset{x}{\text{minimize}} \sum_{i\in\delta^-(d)} \sum_{k=1}^{K} \sum_{t=0}^{T} t \cdot x_{idk}(t)$$

subject to

$$\sum_{j\in\delta^+(i)} \sum_{k=1}^{K} \sum_{t=0}^{T} x_{ijk}(t) = r(i) + \sum_{j\in\delta^-(i)} \sum_{k=1}^{K} \sum_{t=0}^{T} x_{jik}(t), \quad \forall i \in S \tag{1}$$

$$\sum_{j\in\delta^-(d)} \sum_{k=1}^{K} \sum_{t=0}^{t+w_k\times\lambda_{jd}\leq T} x_{jdk}(t) = \sum_{i\in S} r(i) \tag{2}$$

$$\sum_{j\in\delta^-(i)} \sum_{k=1}^{K} x_{jik}(t - w_k \times \lambda_{ji}) - \sum_{j\in\delta^+(i)} \sum_{k=1}^{K} x_{ijk}(t) + y_i(t-1) = y_i(t), \quad \forall i \in N\backslash\{d\},\ \forall t > 0 \tag{3}$$

$$(k-1) \times \mu_{ij} \times z_{ijk}(t) \leq x_{ijk}(t) \leq k \times \mu_{ij} \times z_{ijk}(t), \quad \forall i,j,k,t \tag{4}$$

$$z_{ijk}(t) \times \sum_{m=1}^{k-1} \sum_{n=t}^{t+(w_k-w_m)\lambda_{ij}-1} z_{ijm}(n) = 0, \quad \forall i,j,k,t \tag{5}$$

$$\sum_{k=1}^{K} z_{ijk}(t) \leq 1, \quad \forall i,j,t \tag{6}$$

$$0 \leq y_i(t) \leq c_i, \quad \forall i,t \tag{7}$$

$$z_{ijk}(t) \in \{0,1\}, \quad \forall i,j,k,t \tag{8}$$

Constraint (1) ensures that the number of evacuees who have left a node until time T equals the sum of the initial number of evacuees in the node and the number of evacuees who have entered the node until time T. Constraint (2) ensures that all evacuees arrive at the super sink node within the final evacuation time T. Constraints (1) and (2) are called as the source and sink balancing constraints, respectively. Unlike the formulation in [13], where source nodes cannot appear in the middle of the paths starting from other source nodes, all source nodes can show up in the paths from other source nodes. Constraint (1) constructs the paths including source nodes. Constraint (3), the flow conservation constraint, forces the difference between inflow and outflow at time t in a node, except the super sink node, to match the difference between the number of evacuees staying in a node at time t and time $t-1$. Constraint (4) ensures that the number of evacuees on the arc from node i to node j with class k at time t cannot exceed k times the unit capacity of the arcs. That is, $x_{ijk}(t)$ does not exceed its arc capacity. Constraint (5) avoids the FIFO violation shown in Fig. 4. Constraint (6) guarantees that evacuees use only one class of arc at a time. Constraint (7) ensures that the number of evacuees in each source node cannot exceed the node capacity. Constraint (8) defines $z_{ijk}(t)$ as a binary variable.

$$a_{ijk}(t) \leq \sum_{m=1}^{k-1} \sum_{n=t}^{t+(w_k-w_m)\lambda_{ij}-1} z_{ijm}(n), \quad \forall i,j,k,t \tag{9}$$

$$a_{ijk}(t) \geq \left\lfloor \frac{\sum_{m=1}^{k-1} \sum_{n=t}^{t+(w_k-w_m)\lambda_{ij}-1} z_{ijm}(n) - 1}{(k-1)w_k - (w_1 + w_2 + \ldots + w_{k-1})} + 1 \right\rfloor + z_{ijk}(t) - 1, \quad \forall i,j,k,t$$

$$a_{ijk}(t) \leq z_{ijk}(t), \quad \forall i,j,k,t \tag{10}$$

$$a_{ijk}(t) \in \{0,1\}, \quad \forall i,j,k,t \tag{11}$$

Noting that Constraint (5) is quadratic, we linearize it by introducing new binary decision variables $a_{ijk}(t)$ for all i, j, k, t as in [17], [18], and [19]. The result is that Constraint (5) is transformed into Constraints (9) – (11). Constraints (9) – (11), however, significantly worsen computational performance, which motivates us to develop an efficient heuristic algorithm described in Section V.

B. MILP MODEL WITH SINGLE-CLASS ARCS

The MILP model for the MCTE network is solvable only for small-size network cases, so it would be good to have upper bounds for the evacuation time before moving to the heuristic algorithm. The solution of the MILP model for the MCTE network tends to use a higher class of arcs when there are many evacuees and a lower class of arcs when there are fewer evacuees in the network. We consider networks with only single-class arcs and use them as bounds for the performance of our heuristic algorithm. For example, if the MCTE network has three classes of arcs, we build three MILP models for each class, by removing Constraints (5), (6), (8), and binary

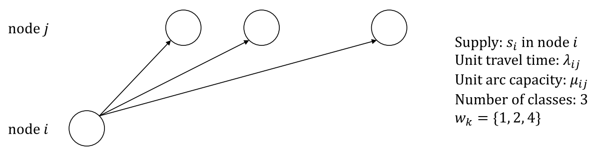


FIGURE 6. Two nodes example of using multiple classes.

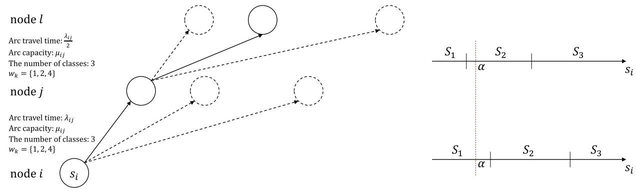


FIGURE 7. Using different class arc in the same evacuation path.

variables $z_{ijk}(t)$. Since the single-class MILPs are solvable for relatively large-size networks, we use them as references for the performance of our heuristic algorithm. Section VI discusses our numerical examples and results. We formulate the MILP model with single-class networks as follows:

$$\underset{x}{\text{minimize}} \sum_{i\in\delta^-(d)} \sum_{t=0}^{T} t \times x_{id}(t)$$

subject to

$$\sum_{j\in\delta^+(i)} \sum_{t=0}^{T} x_{ij}(t) = r(i) + \sum_{j\in\delta^-(i)} \sum_{t=0}^{T} x_{ji}(t), \quad \forall i \in S$$

$$\sum_{j\in\delta^-(d)} \sum_{t=0}^{t+\lambda_{jd}\le T} x_{jd}(t) = \sum_{i\in S} r(i)$$

$$\sum_{j\in\delta^-(i)} x_{ji}(t-\lambda_{ji}) - \sum_{j\in\delta^+(i)} x_{ij}(t) + y_i(t-1) = y_i(t),$$

$$\forall i \in N\backslash(\{d\}), t$$

$$0 \le y_i(t) \le c_i, \quad \forall i, t$$

$$0 \le x_{ij}(t) \le \mu_{ij}, \quad \forall i, j, t$$

V. HEURISTIC ALGORITHM

A rapidly unfolding emergency situation requires prompt decision making. In this section, we describe our heuristic algorithm for finding evacuation paths. The basic idea of our heuristic algorithm is to create paths by choosing an *effective class* of arcs from source nodes to sink nodes. An *effective class* arc from node i to j is an arc through which evacuees can move from node i to node j faster than move through any other classes of arcs. The effective class between two nodes depends on the number of evacuees in the tail node and the travel time/capacity of each class arc. For better understanding, we start with an example, as illustrated in Fig. 6.

For two nodes i and j with three classes of arcs in Fig. 6, we can define set S_i^k whose elements are the values of the number of evacuees in node i when the class k arc is effective. When the number of evacuees is less than $2\lambda_{ij}\mu_{ij}$, the class 1 arc is effective. S_i^2 and S_i^3 can be obtained for class 2 and class 3 arcs as follows:

The set for class 1 arc (S_i^1) is

$$\begin{aligned} S_i^1 &= \left\{ s_i \middle| \left\lfloor \frac{s_i}{\mu_{ij}} \right\rfloor + \lambda_{ij} \le \left\lfloor \frac{s_i}{2\mu_{ij}} \right\rfloor + 2\lambda_{ij} \right\} \\ &\quad \cap \left\{ s_i \middle| \left\lfloor \frac{s_i}{\mu_{ij}} \right\rfloor + \lambda \le \left\lfloor \frac{s_i}{3\mu_{ij}} \right\rfloor + 4\lambda_{ij} \right\} \\ &= \{ s_i | s_i \le 2\lambda_{ij}\mu_{ij} \}. \end{aligned}$$

The set for class 2 arc (S_i^2) is

$$\begin{aligned} S_i^2 &= \left\{ s_i \middle| \left\lfloor \frac{s_i}{2\mu_{ij}} \right\rfloor + 2\lambda_{ij} \le \left\lfloor \frac{s_i}{\mu_{ij}} \right\rfloor + \lambda_{ij} \right\} \\ &\quad \cap \left\{ s_i \middle| \left\lfloor \frac{s_i}{2\mu_{ij}} \right\rfloor + 2\lambda_{ij} \le \left\lfloor \frac{s_i}{3\mu_{ij}} \right\rfloor + 4\lambda_{ij} \right\} \\ &= \{ s_i | 2\lambda_{ij}\mu_{ij} \le s_i \le 12\lambda_{ij}\mu_{ij} \}. \end{aligned}$$

The set for class 3 arc (S_i^3) is

$$\begin{aligned} S_i^3 &= \left\{ s_i \middle| \left\lfloor \frac{s_i}{3\mu_{ij}} \right\rfloor + 4\lambda_{ij} \le \left\lfloor \frac{s_i}{\mu_{ij}} \right\rfloor + \lambda_{ij} \right\} \\ &\quad \cap \left\{ s_i \middle| \left\lfloor \frac{s_i}{3\mu_{ij}} \right\rfloor + 4\lambda_{ij} \le \left\lfloor \frac{s_i}{2\mu_{ij}} \right\rfloor + 2\lambda_{ij} \right\} \\ &= \{ s_i | s_i \ge 12\lambda_{ij}\mu_{ij} \}. \end{aligned}$$

We can derive S_i^k for general cases with N classes as follows:

$$S_i^k = \left\{ s_i \middle| \bigcap_{l=1}^{N} \left(\left\lfloor \frac{s_i}{k \cdot \mu_{ij}} \right\rfloor + w_k \cdot \lambda_{ij} \le \left\lfloor \frac{s_i}{l \cdot \mu_{ij}} \right\rfloor + w_l \cdot \lambda_{ij} \right) \right\}$$

We choose effective class k when the tail node i has the number of evacuees that belongs to set S_i^k. We note that as the number of evacuees increases, using higher class arcs becomes effective.

One can construct a path by iterating the process of selecting the effective arc until we find exit nodes (see Fig. 7). Suppose that the number of remaining evacuees in node i (s_i) is α. The effective class of arcs from node i to node j is class 1, but the effective class from node j to node l is class 2.

Selecting an effective arc in every step can help construct better paths, but requires more computational effort. We want our algorithm to run faster, not compromising the solution quality a lot. We find a clue for improving the speed of the algorithm from the solutions of the MILP model for the MCTE network. When a source node has many evacuees (congested), the solution tends to choose higher class arcs to push more evacuees to exit nodes using large capacity. On the other hands, when there are few evacuees, the solution selects lower class arcs utilizing the short travel times. So, the solution tends to choose a single class (either the highest or the lowest) throughout a path when the source nodes are congested or almost empty. From this observation, we design our algorithm to choose only one class for each path and assign evacuees to it.

Consider a three-node example in Fig. 8. Suppose two base arcs have same travel times, capacities and a weight set for classes ($w_k = \{1, 2, 4\}, k = 1, 2, 3$). Our algorithm

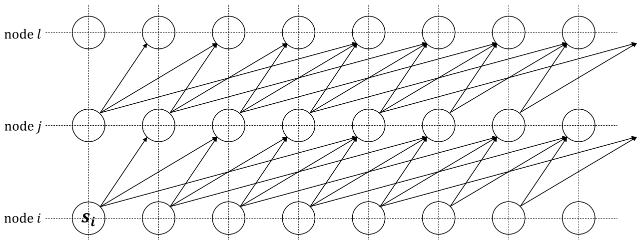


FIGURE 8. Three nodes example of choosing class of arcs.

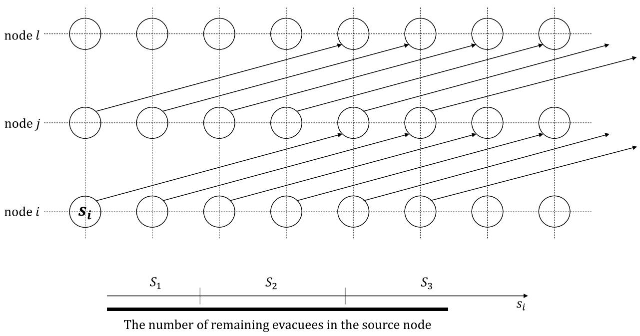


FIGURE 9. Three nodes example of choosing class 3 arcs.

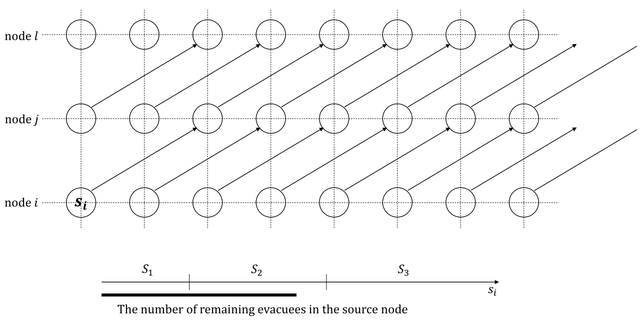


FIGURE 10. Three nodes example of choosing class 2 arcs.

chooses the effective class from S_i^k as explained above. Fig. 9 and Fig. 10 show two cases where the effective classes are 3 and 2 respectively. When the number of remaining evacuees in node i is in S_i^3, our heuristic algorithm finds evacuation paths using only class 3 arcs. As the number of remaining evacuees in node i decreases, it will enter S_i^2, and finally will be in S_i^1. Our heuristic algorithm will use class 2 arcs and eventually use class 1 arcs until the evacuation is completed.

A. STRUCTURE OF HEURISTIC ALGORITHM

Algorithm 1 shows the overall structure of our heuristic algorithm. The algorithm consists of two parts: generating networks (Algorithm 2 in Section V-B) and finding evacuation paths (Algorithm 3 in Section V-C). While the main algorithm is finding evacuation paths (Algorithm 3), we incorporate how we build an MCTE network (Algorithm 2) as a part of the algorithm.

Let T_{max} be the initial number of time windows. We do not know when the last evacuee will exit the building before executing the algorithm. So, we set the T_{max} to be sufficiently large and if the evacuation is not done at time T_{max}, we increase the number of time windows by T_{inc} at a time. Building network information is used when generating networks (Algorithm 2).

Algorithm 1 Structure of Heuristic Algorithm

Input:
Max Time(T_{max}): Initial number of time windows
Increment Time(T_{inc}): Number of additional time windows
Extension counts(E): Number of network extensions
Extension limit(L): Maximum number of network extensions
Building Network Information: Information about spaces, evacuees, floors, stairs, hallways, etc
Output:
Evacuation plans

```
Run Algorithm2(Building Network Information)
Run Algorithm3(Generated MCTE network)

Extension counts(E) ← 0
while There are remaining evacuees in network do
    if E ≤ L then
        Extend MCTE network by T_inc
        E ← E + 1, T_max ← T_max + T_inc
        Run Algorithm3(Extended MCTE network)
    else
        Stop finding Evacuation paths
```

Using the input data, Algorithm 2 determines the details of networks such as capacities of nodes and arcs, travel times of arcs, the number of evacuees in source nodes, etc. Algorithm 2 initially creates T_{max} time windows, copies all nodes on the time windows, and connects all arcs among all nodes. The details of Algorithm 2 are in Section V-B.

After generating the MCTE network, Algorithm 3 finds evacuation paths. If there are still remaining evacuees at time T_{max}, the algorithm will extend the MCTE network by T_{inc}. Then, the algorithm finds more evacuation paths in the extended time windows. We, however, should carefully take care of special cases which may make the algorithm fail; when some nodes are physically isolated by fire, smoke, etc., the time extension cannot resolve the problem and the algorithm never stops. To prevent such situations, we set a limit on the number of times of network extensions (L). If some evacuees still remain in the network after L extensions, the algorithm will stop finding evacuation paths. We explain the details of Algorithm 3 in Section V-C.

B. GENERATING MCTE NETWORKS

We use building information generating MCTE networks with node capacities, node supply (the initial number of evacuees in a node), tail/head nodes (multi-class arcs), arc capacities, and arc travel times.

Algorithm 2 Generating MCTE Networks

Input:
Building Network Information(Building plans including footage of spaces, number of evacuees, widths, and lengths of hallways and stairs
$\mathcal{S}=\{s_i|i=1,2,\ldots,S\}$, $\mathcal{E}=\{e_i|i=1,2,\ldots,E\}$: Sets of source and exit nodes
$\mathcal{N}=\{n_i|i=1,2,\ldots,N\}$, $\mathcal{A}=\{a_i|i=1,2,\ldots,A\}$: Set of all nodes and arcs
$\mathcal{W}=\{w_k|k=1,2,\ldots,K\}$: Weights of class i arcs' travel time
T_{max}, T_{inc}: Initial max time and Increment time
Output:
Multi-class Time-expanded network(MCTE network)

```
for each time window t ≤ T_max do
    Make copies of all nodes in each time window t
    for each node n_i ∈ N do
        if t = 0 and there are evacuees in n_i then
            n_i is source node
        else if t ≥ 1 and n_i has infinite capacity then
            n_i is exit node
        else
            n_i is normal node
for each arc a_i ∈ A do
    for each class k do
        tail ← a_i's tail
        head ← a_i's head + a_i's travel time × w_k
        while head is in time window t ≤ T_max do
            Connect arc between tail and head
            arc capacity is (a_i's arc capacity) × k
            tail ← same node in next time window
            head ← same node in next time window
```

At the beginning of Algorithm 2, we make copies of all nodes for each time window. Each node has information on its capacity, supply, and type (source/exit/normal). The type of a node is determined based on the following criteria: if at least one evacuee is in the node at time 0, it is a source node. If it has infinite capacity, it is an exit node. Otherwise, it is a normal node. After setting all nodes in the MCTE network, we connect them by using the arc information (tail nodes, head nodes, capacities, travel times and classes of arc).

The MCTE network has transit arcs that connect two different nodes, and hold-over arcs that represent evacuees' waiting in the same node for avoiding congestion. Note that hold-over arcs' capacities and their tail or head node capacities are the same. In an MCTE network, individual evacuees use a transit arc or a hold-over arc to move to the next time window. Evacuees using the hold-over arc just stay in the same node.

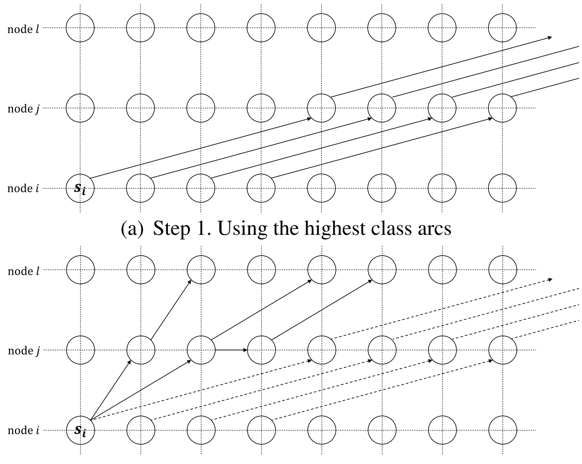
C. FINDING EVACUATION PATHS

Algorithm 3 shows how we construct evacuation paths and allocate evacuees to them. Due to the time-expanded property, all nodes are duplicated in each time window and are connected by arcs according to travel times. In addition, due to the multi-class property, the number of arcs is several times higher than the number of physical connections. To reduce computation time, we propose a sequential node activating/deactivating technique by restricting our search space. We arbitrarily choose the first activated node from the source nodes at time 0; we also tried to choose the source node having the largest initial supply, but the performance is almost the same. We then sequentially activate nodes which are directly connected to activated nodes until activating one of the exit nodes. When one of the exit nodes is activated, we can construct an evacuation path by backtracking from the activated exit nodes to the source node through activated nodes. To prevent too many activated nodes for search efficiency, we adopt a greedy approach choosing *effective* class arcs (i.e., higher class arcs) so that more evacuees can move at a time. Recall that as the number of evacuees (s_i) in node i increases, it belongs to a higher effective class set, S_i^k. That is, we choose higher class arcs when the number of evacuees in the source node is large.

Algorithm 3 Finding Evacuation Paths

Input:
MCTE network generated by Algorithm 2
$\mathcal{S}=\{s_i|i=1,2,\ldots,S\}$: Set of source nodes
$\delta_k^+(n)=\{d_i|i=1,2,\ldots,N\}$: Set of head nodes of node n connected by class k arcs
$\mathcal{I}=\{I_k|k=1,2,\ldots,K\}$: Minimum number of evacuees for using class k arcs
Output:
Evacuation Paths

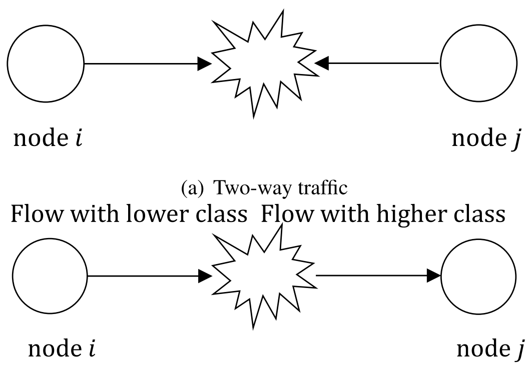
```
for each class k do
    for each source node s_i ∈ S do
        Deactivate all nodes in MCTE Network
    while Number of evacuees in s_i > I_k do
        Activate all nodes in δ_k^+(s_i)
        N* = {n_i^* | i = 1, 2, 3...}: set of activated nodes
        while there are no exit nodes in N* do
            for each node n_i^* ∈ N* do
                Activate all nodes in δ_k^+(n_i^*)
                if there are exit nodes in δ_k^+(n_i^*) then
                    break
        Select an evacuation path between s_i and exit node in δ_k^+(n_i^*)
        Update arcs' and nodes' capacities in the evacuation path
        Reduce number of remaining evacuees in s_i
if there are remaining evacuees in MCTE network then
    Extend MCTE network; T_max ← T_max + T_inc
    find evacuation paths in extended MCTE network
```



(a) Step 1. Using the highest class arcs

(b) Step 2. Using lower class arcs

FIGURE 11. Steps of finding evacuation paths.



(a) Two-way traffic

(b) FIFO violation

FIGURE 12. Two kinds of collision in arcs.

We use two techniques, node activating/deactivating and effective classes, to construct evacuation paths. We find evacuation paths using the effective class arcs for each source node depending on the number of remaining evacuees. As shown in Fig. 11(a), we construct the evacuation paths using effective class arcs. As the evacuation progresses, the number of remaining evacuees decreases, and the effective class becomes a lower class as shown in Fig. 11(b). Once evacuation from a source node is done, we iterate the same procedure for other source nodes until all evacuees exit.

We should avoid situations including simultaneous two-way traffic by evacuees moving in opposite directions in an arc and FIFO violations as done in the MILP model for the MCTE network. Fig. 12 illustrates both situations. Two-way traffic occurs when some evacuees move from node i to node j at the same time while other evacuees move from node j to node i. A FIFO violation occurs when evacuees who use the lower class arc (shorter travel time with smaller capacity) pass those moving through the higher class arc (longer travel time with larger capacity). To solve the two-way traffic problem, when node i sends evacuees to node j at time t, we set the capacity of the arc from node j to node i to zero. To avoid the FIFO violation, we choose the class k arc connecting node i at time t_0 and node j at time t_1, and set the capacity of the arcs that connect node i later than time t_0 and node j earlier than time t_1 to zero.

VI. NUMERICAL RESULTS

In this section, we show several numerical results comparing the proposed heuristic algorithm with the MILP models: the MCTE network and two one-class networks. To the best of our knowledge, we are the first one introducing the concept of the MCTE network for different congestion levels and their effects; it is hard to find appropriate existing algorithms for benchmarking other than the MILP models.

We use the final evacuation time as a primary performance measure for the quality of generated paths and the solution time as a secondary performance measure for quick decision making. We write code using the C++ interface of GUROBI 7.52 for solving the MILP models and run it on a Microsoft Windows Server 2012 R2 with Inter(R) Xeon(R) CPU E3-1230 V2 @ 3.30GHz, 16.0GB RAM and Single thread. In large-size networks, however, the MILP model for the MCTE network is not solvable, so we compare the solution of our heuristic algorithm to the solutions of the MILP models with single-class arcs; see Section IV-B for an explanation.

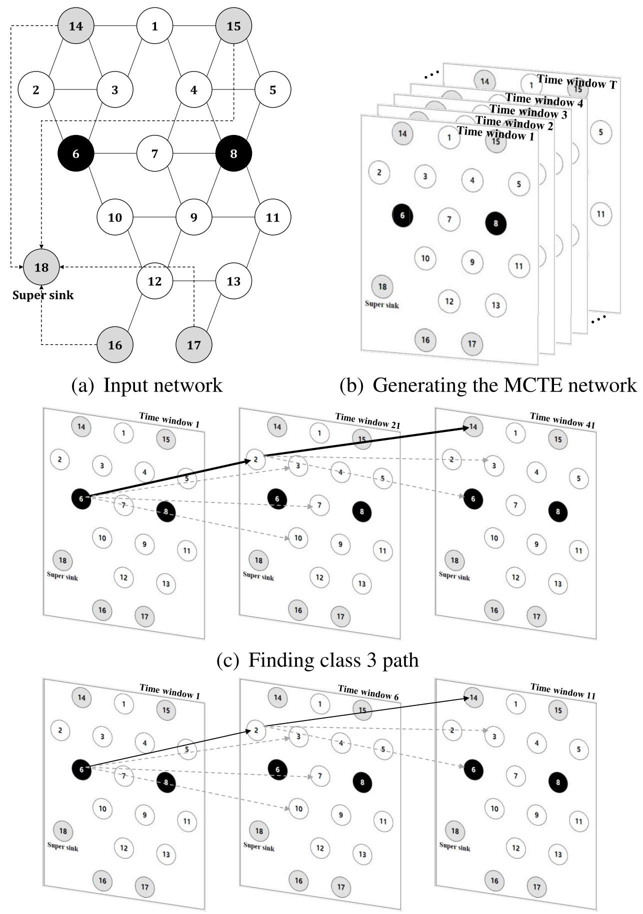
We consider four networks for numerical experiments. The first network is a small-size artificial network we make up to compare the performance of our algorithm with the optimal solution from the MILP model for the MCTE network. Three others are a real multiplex cinema network, a subway station network, and a complex shopping mall network.

A. SMALL-SIZE ARTIFICIAL NETWORK

We generated a network which consists of 18 nodes and 54 arcs. There are four exit nodes and one super sink node. We consider two settings for the multi-class property: Setting 1 has two classes ($K = 2$, $w_k = \{1, 2\}$) and Setting 2 has three classes ($K = 3$, $w_k = \{1, 2, 4\}$). For Setting 1, when we set the lower class arc's capacity to be 3 and the base travel time to be 5, the higher class arc's capacity is 6 (base capacity $\times\ k = 3 \times 2$) and the travel time is 10 (base travel time $\times\ w_k = 5 \times 2$).

Fig. 13 shows the process of the proposed heuristic algorithm to find evacuation paths. Fig. 13(a) is one of the input data of Algorithm 2. We generate the MCTE network for the small-size artificial network in Fig. 13(b) based on the input network in Fig. 13(a). In Algorithm 3, we check the number of evacuees in each source node and choose the effective class of arcs to find evacuation paths. Fig. 13(c) and 13(d) show examples of evacuation paths using class 3 arcs and class 1 arcs respectively. The arc between node 6 and node 2 in Fig. 13(c) has 3 times more capacity and 4 times longer travel time than the arc in Fig. 13(d).

This experiment uses the MILP models for the MCTE network and the single-class MILP models as performance references. There are two single-class MILP models; one uses class 1 arcs and the other uses class 2 arcs. The network

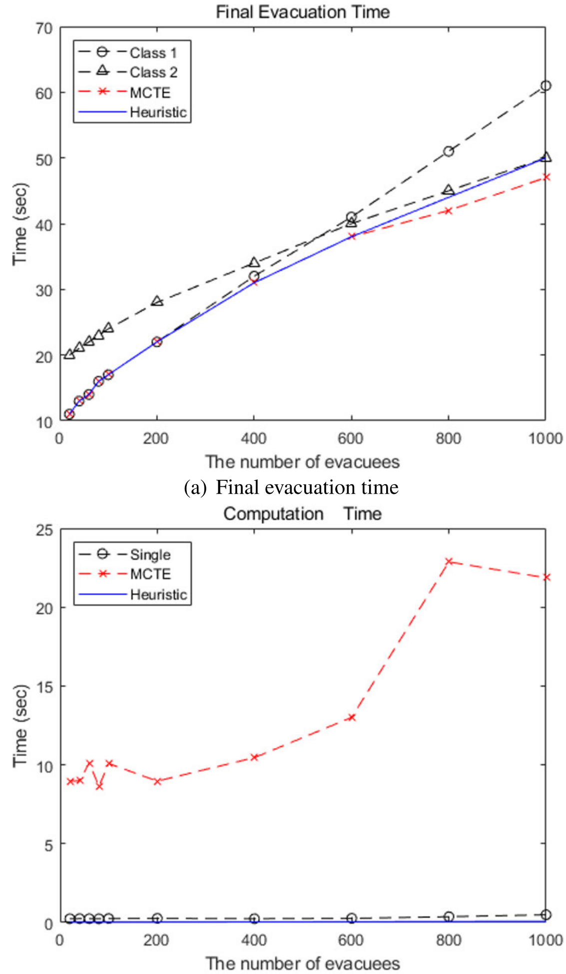


(a) Input network (b) Generating the MCTE network

(c) Finding class 3 path

(d) Finding class 1 path

FIGURE 13. **The process of heuristic algorithm.**



(a) Final evacuation time

(b) Computation time

FIGURE 14. **The artificial network with 2 classes.**

using class 1 arcs represents the strategy that evacuates a small number of evacuees from one node to another with shorter travel time. The network using class 2 arcs represents the strategy that evacuates a large number from one node to another with longer travel time. Fig. 14(a) shows the final evacuation time using only class 1 arcs, class 2 arcs, the MILP for the MCTE network, and the heuristic algorithm. The graphs labeled Class 1 (class 1 arc) and Class 2 (class 2 arc) show the solutions of the MILP models for single-class arcs. The graph labeled MCTE shows the solution of the MILP model for the MCTE network. The solid line labeled Heuristic comes from the heuristic algorithm. Fig. 14 shows that the final evacuation time of the MILP model using class 1 arcs is the fastest when the number of evacuees in the network is smaller than 200. The results of the MILP model for the MCTE network and the heuristic algorithm are the same as the MILP model using class 1 arcs. When the number of evacuees exceeds 600, only the MILP model for the MCTE network guarantees optimality. Even though our heuristic algorithm cannot guarantee optimality, it is better than the two MILP models for single-class arcs. The MILP model for the MCTE network can always use optimal arcs for all time windows. And, evacuation paths from the MILP model for the MCTE network can consist of a non-homogeneous class of arcs. However, our heuristic algorithm chooses arcs considering the number of remaining evacuees in each source node and chooses only one kind of class arcs in each path.

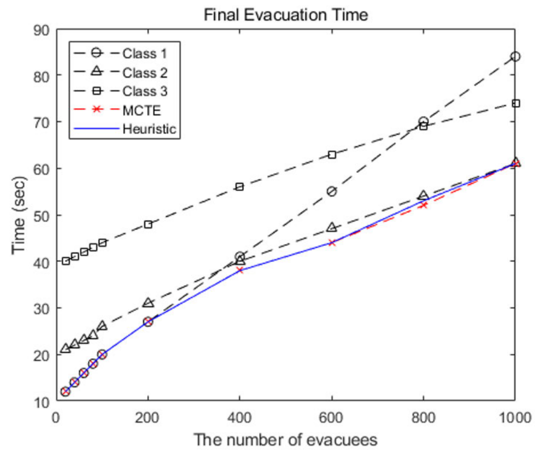
Fig. 14(b) shows the computation time of the two simple MILP models. The computation time of the MILP model for the MCTE network depends on the number of evacuees ranges from 8.6 to 23 seconds. The artificial network is indeed very small, so our heuristic algorithm takes less than 0.1 seconds. Table 2 summarizes the average computation times.

In Setting 2, we use the three single-class MILP models corresponding to class 1, 2, and 3 arcs. We vary the number of evacuees in the network from 20 to 4000. Since the number of classes is now three, the computation cost is higher than Setting 1. When the number of evacuees is larger than 1000, the solver cannot solve the MILP model for the MCTE network.

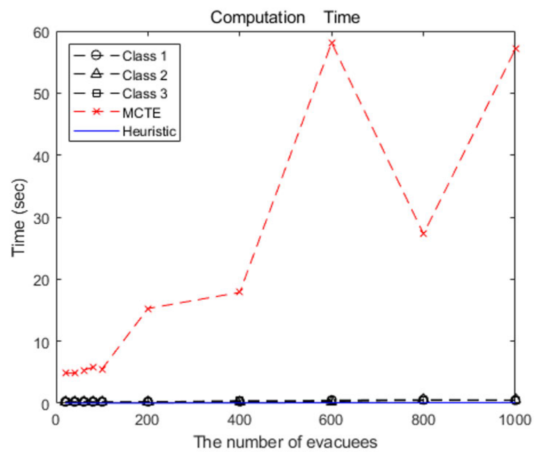
Fig. 15 shows that the proposed heuristic algorithm obtains a near-optimal solution close to that of the MILP model for the MCTE network when the number of evacuees in the

TABLE 2. Computation times (sec) for the artificial network with 2 classes.

Number of Evacuees	MILP-single class (Average)	MILP-MCTE	Heuristic
20	0.238	8.964	0.018
40	0.246	8.990	0.019
60	0.241	10.114	0.019
80	0.237	8.630	0.019
100	0.251	10.084	0.021
200	0.261	8.965	0.024
400	0.241	10.458	0.037
600	0.263	13.011	0.047
800	0.368	22.868	0.060
1000	0.492	21.856	0.069



(a) Final evacuation time



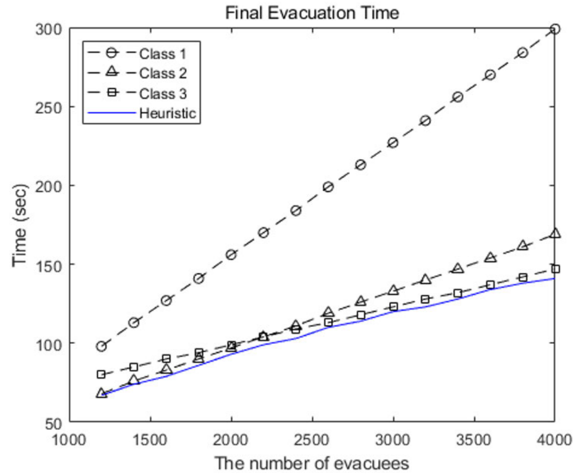
(b) Computation time

FIGURE 15. The artificial network with 3 classes and ≤ 1,000 evacuees.

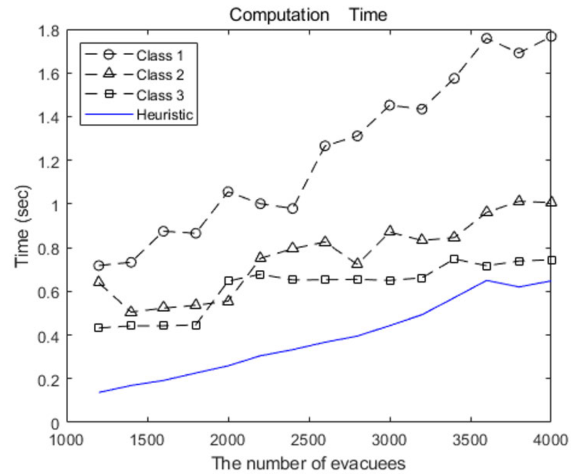
network ranges between 20 and 1000. When the number of evacuees is larger than 1000, we only compare the solution of proposed algorithm to the solutions of the three single-class MILP models.

Fig. 16 shows that it is preferable to use higher class arcs when there are more evacuees. Our heuristic algorithm always shows better final evacuation times and faster computation times than the single-class MILP models regardless of the number of evacuees. Fig. 15(b) shows that the MILP model for the MCTE network needs at least 4.846 seconds for the small-population network, whereas the computation times of the single-class MILP models and our heuristic algorithm need less than 0.5 seconds.

Table 3 lists the computation times shown in Fig. 16. In summary, we observe that our heuristic algorithm obtains near-optimal solutions for small-population networks in less time. Our heuristic algorithm always outperforms the three single-class MILP models for large-population networks.



(a) Final evacuation time



(b) Computation time

FIGURE 16. The artificial network with 3 classes and ≥ 1,000 evacuees.

TABLE 3. Computation times (sec) for the artificial network with 3 classes more than 1,000 evacuees.

Number of Evacuees	MILP-single class (Average)	MILP-MCTE	Heuristic
20	0.230	4.846	0.016
40	0.230	4.856	0.018
60	0.244	5.319	0.018
80	0.233	5.796	0.019
100	0.232	5.498	0.020
200	0.241	15.269	0.028
400	0.315	17.865	0.051
600	0.390	58.103	0.064
800	0.494	27.268	0.079
1000	0.488	57.067	0.108

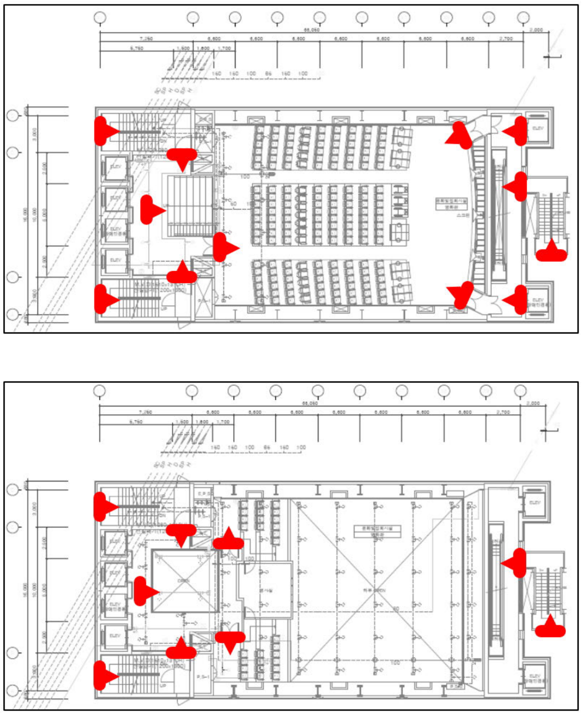


FIGURE 17. **A floor plan of a multiplex cinema.**

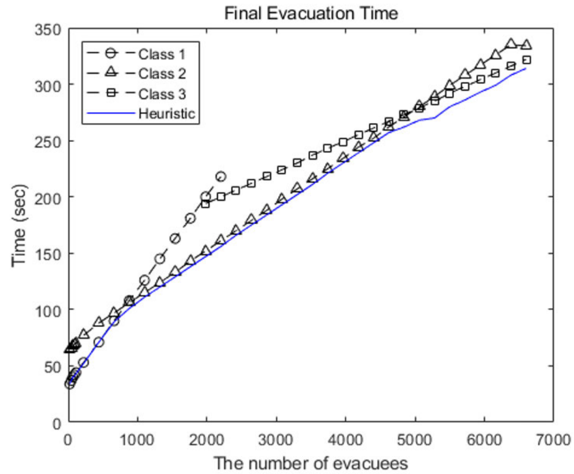
B. MULTIPLEX CINEMA NETWORK

The second network is a real multiplex cinema network in Seoul, Korea. Fig. 17 shows the floor plan. The red boxes in Fig. 17 represent the places where we set the nodes. The network consists of 99 nodes and 255 arcs. Among the 99 nodes, there are 11 potential source nodes, three exit nodes, and one super sink node. For the multi-class property, we use 3 classes ($K = 3$ and $w_k = \{1, 2, 4\}$) and vary the number of evacuees between 22 and 6600. Since the MILP model for the MCTE network is unsolvable for all cases, we use the three single-class MILP models for the performance comparison.

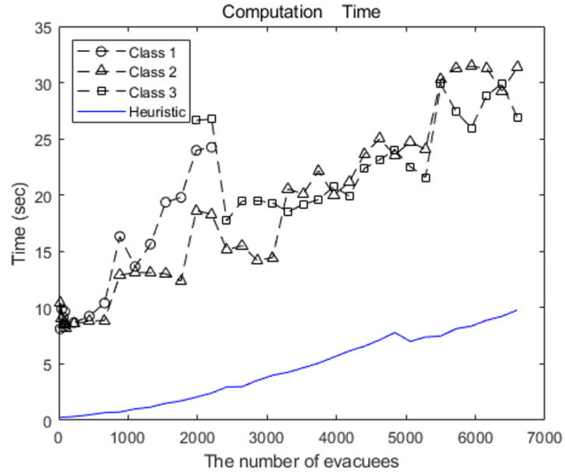
Fig. 18 shows that the single-class MILP model with class 1 arcs is the most efficient when there are fewer than 660 evacuees, the model with class 2 arcs is the most efficient when there are 880 to 4840 evacuees, and the model with class 3 arcs is the most efficient when there are more than 4840 evacuees. Our heuristic algorithm, which can use all classes of arcs, obtains a better solution than the single-class MILP models. Fig. 18(b) shows that the single-class MILP models need 8 seconds compared to our heuristic algorithm that only needs 10 seconds at most. Table 4 lists the computation times for the multiplex cinema network.

C. SUBWAY STATION NETWORK

The third network is a real subway station. We choose a subway station network in Busan, Korea equipped with an IoT-based evacuation system using the previous version of our evacuation algorithm not having the multi-class property.



(a) Final evacuation time



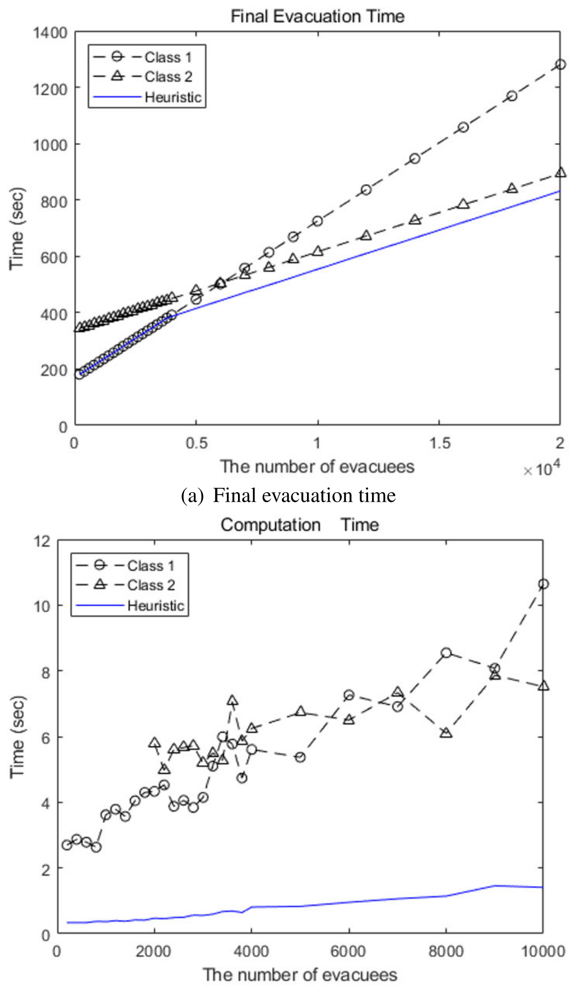
(b) Computation time

FIGURE 18. **The multiplex cinema network with 3 classes.**

TABLE 4. **Computation times (sec) for multiplex cinema.**

Number of Evacuees	MILP single class (Average)	Heuristic	Number of Evacuees	MILP single class (Average)	Heuristic
22	9.269	0.253	3080	16.881	4.000
44	9.466	0.266	3300	19.525	4.259
66	8.827	0.275	3520	19.650	4.660
88	9.128	0.277	3740	20.882	5.072
110	8.305	0.284	3960	20.391	5.609
220	8.622	0.327	4180	20.533	6.149
440	9.017	0.476	4400	23.015	6.569
660	9.634	0.673	4620	24.075	7.138
880	14.604	0.728	4840	23.771	7.789
1100	13.395	1.005	5060	23.644	7.002
1320	14.368	1.164	5280	22.817	7.393
1540	16.184	1.503	5500	30.066	7.477
1760	16.090	1.718	5720	29.362	8.128
1980	23.072	2.053	5940	28.697	8.360
2200	23.114	2.415	6160	30.038	8.883
2420	16.436	2.958	6380	29.590	9.233
2640	17.485	2.982	6600	29.125	9.785
2860	16.853	3.549			

The network consists of 51 normal nodes, 6 exit nodes, and 162 arcs. Most of the arcs have a large capacity and long travel time, so the highest class does not seem necessary. When we run our proposed heuristic algorithm with 3 classes,



(a) Final evacuation time

(b) Computation time

FIGURE 19. The subway station network with 3 classes.

class 3 arcs are not selected. Therefore, we only compare two single-class MILP models and our heuristic algorithm. For the multi-class property, we set $K = 2$, $w_k = \{1, 2\}$, and increase the number of evacuees from 200 to 20000.

When the network has a small number of evacuees, the MILP model for the MCTE network is solvable, but it needs a very long time to solve, and the results are the same as those of the single-class MILP model with class 1 arcs and the heuristic algorithm. Therefore, in Fig. 19, we only plot the results of two single-class MILP models and our heuristic algorithm. Fig. 19(a) shows the final evacuation times. Similar to the multiplex cinema network case, our heuristic algorithm outperforms the two single-class MILP models. The real subway station network has a small number of wide corridors and stairs, so the number of paths for evacuees is smaller than other real-world cases. Fig. 19(b) and Table 5 show the computation times of single-class MILP and our heuristic algorithm.

D. COMPLEX SHOPPING MALL NETWORK

The last network is a real shopping mall in Busan, Korea, providing entertainment, dining, etc. When malls add attractions other than shopping, the sites add complexity, so visitors may be unable to locate emergency exits. We choose an eleven-story mall consists of 229 nodes and 709 arcs. The network has 11 exit nodes on the first floor and a basement. We use 3 classes for the multi-class property with $K = 3$ and $w_k = \{1, 2, 4\}$. Since the size is larger than the three other cases, the computation times are significantly longer. We increase the number of evacuees in the network from 200 to 10000.

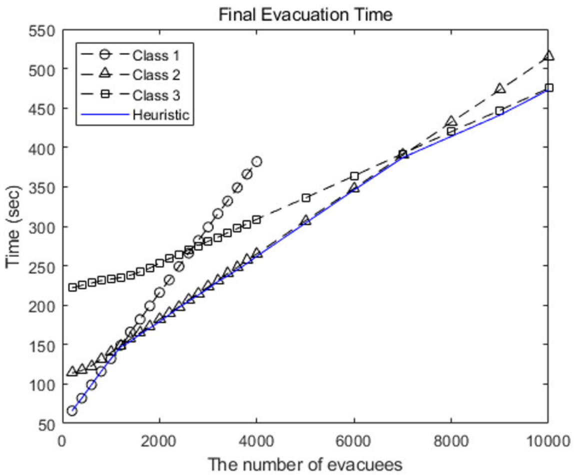
Fig. 20(a) shows the final evacuation times for the three single-class MILP models and our heuristic algorithm. When the number of evacuees is fewer than 1200, the single-class MILP model with class 1 arcs is the most efficient, and when the number of evacuees is larger than 8000, the single-class MILP model with class 3 arcs is the most efficient. When the number of evacuees is larger than 8000, our heuristic algorithm uses all classes of arcs and finds a better or competitive

TABLE 5. Computation times (sec) for a subway station network.

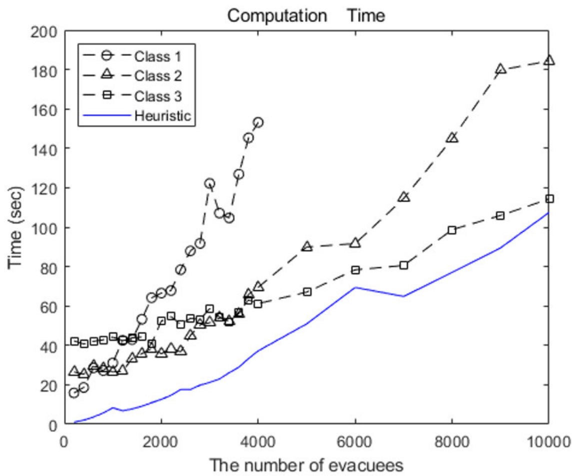
Number of Evacuees	MILP single class (Average)	Heuristic	Number of Evacuees	MILP single class (Average)	Heuristic
200	2.697	0.335	2800	4.779	0.565
400	2.870	0.337	3000	4.670	0.555
600	2.785	0.336	3200	5.299	0.593
800	2.631	0.375	3400	5.633	0.672
1000	3.618	0.366	3600	6.417	0.689
1200	3.786	0.396	3800	5.298	0.644
1400	3.566	0.376	4000	5.925	0.811
1600	4.046	0.418	5000	6.048	0.831
1800	4.298	0.411	6000	6.883	0.955
2000	5.058	0.470	7000	7.121	1.063
2200	4.751	0.458	8000	7.321	1.143
2400	4.746	0.489	9000	7.969	1.458
2600	4.860	0.500	10000	9.079	1.408

TABLE 6. Computation times (sec) for complex shopping mall network.

Number of Evacuees	MILP class 1	MILP class 2	MILP class 3	Heuristic
200	15.873	26.459	42.267	1.024
400	18.639	25.413	40.706	2.048
600	28.548	29.364	42.092	3.672
800	27.223	28.239	42.801	5.637
1000	31.116	26.556	44.192	8.366
1200	42.505	26.796	42.482	6.798
1400	42.931	33.440	43.815	7.789
1600	53.258	35.796	44.249	9.106
1800	64.176	38.363	40.929	10.923
2000	66.501	35.792	52.349	12.578
2200	67.817	38.272	54.612	14.644
2400	78.407	36.683	50.479	17.630
2600	87.921	45.016	53.707	17.612
2800	91.738	50.420	53.137	19.825
3000	122.175	51.462	58.802	21.216
3200	107.141	53.929	54.502	22.950
3400	104.674	52.467	51.663	26.144
3600	126.871	56.100	56.877	28.931
3800	145.304	65.934	63.052	33.331
4000	153.160	69.464	61.220	37.228
5000		89.926	67.027	50.946
6000		91.667	78.347	69.376
7000		114.522	80.555	64.853
8000		144.839	98.747	77.098
9000		179.719	105.938	89.498
10000		184.349	114.336	107.461



(a) Final evacuation time



(b) Computation time

FIGURE 20. The complex shopping mall network with 3 classes.

solution than the single-class MILP models. Fig. 20(b) and Table 6 show the computation times.

VII. CONCLUSION

This paper proposed a new network considering congestion called a multi-class time-expanded (MCTE) network and an efficient heuristic algorithm to solve an evacuation problem under the MCTE network. In the MCTE network, we address congestion in two ways. Multi-class arcs are defined for different travel times and capacities between two nodes depending on the number of evacuees. Time-expanded networks are used for tracking the evacuation process. For that, we first built an MILP model for the MCTE network but found that directly solving the MILP model with commercial solvers is computationally too expensive. We, therefore, developed a heuristic algorithm that reduced search space by restricting classes when constructing a path from source nodes to exit nodes. For the performance comparison, we use additional MILP model not considering the multi-class property.

According to numerical experiments, our heuristic algorithm can obtain near-optimal solutions with much shorter computation time than the MILP models for the MCTE network. An artificial network and the networks of a real multiplex cinema, subway station, and complex shopping mall were used to compare the proposed algorithm's performance against the MILP model for the MCTE networks and the single-class MILP models. The proposed heuristic algorithm provided better or competitive solutions compared to the single-class MILP models and solved the problem much faster compared to the MILP model for the MCTE networks. Furthermore, we could always get feasible solutions in numerical experiments. However, an infeasible solution can be generated in emergencies. When the infeasible solution is generated by the proposed algorithm, we provide information about isolated nodes, locations of evacuees and network configurations to evacuation managers and fire stations.

We suggest several future research directions. Having used a kind of depth-first search approach, considering both depth-first and breadth-first approaches may help construct more efficient paths. We have used the discretized congestion function in generating the MCTE network. One can directly incorporate a continuous congestion function into the mathematical model for more precise congestion control. Another essential research direction is to include human behavior in the mathematical model. In our time-expanded model, we assumed that all evacuees follow the solution of the algorithm, which seems unrealistic in the sense that some evacuees might not follow the instruction and sometimes sending evacuees at a designated rate might be impossible. This limitation is indeed what movement-based approaches, including our model, have in common. Simulation-based approaches, however, can handle human behavior by imposing individual characteristic to each agent in the system, although they can only do what-if analyses. We are investigating how to consider human behavior in our model so that we can solve more realistic problems.

REFERENCES

[1] L. G. Chalmet, R. L. Francis, and P. B. Saunders, "Network models for building evacuation," *Fire Technol.*, vol. 18, no. 1, pp. 90–113, 1982.

[2] L. R. Ford and D. R. Fulkerson, "Constructing maximal dynamic flows from static flows," *Operation Res.*, vol. 6, no. 3, pp. 419–433, 1958.

[3] W. Choi, H. W. Hamacher, and S. Tufekci, "Modeling of building evacuation problems by network flows with side constraints," *Eur. J. Oper. Res.*, vol. 35, no. 1, pp. 98–110, 1988.

[4] H. W. Hamacher and S. Tufekci, "On the use of lexicographic min cost flows in evacuation modeling," *Nav. Res. Log.*, vol. 34, no. 4, pp. 487–503, 1987.

[5] B. Hoppe and V. Tardos, "The quickest transshipment problem," *Math. Oper. Res.*, vol. 25, no. 1, pp. 36–62, 2000.

[6] Q. Lu, Y. Huang, and S. Shekhar, "Evacuation planning: A capacity constrained routing approach," in *Proc. 1st NSF/NIJ Symp. Intell. Secur. Inform.*, H. Chen, R. Miranda, D. D. Zeng, C. Demchak, J. Schroeder, and T. Madhusudan, Eds. Berlin, Germany: Springer, 2003, pp. 111–125.

[7] Q. Lu, B. George, and S. Shekhar, "Capacity constrained routing algorithms for evacuation planning: A summary of results," in *Proc. 9th Int. Symp. Spatial Temporal Databases (SSTD)*, C. B. Medeiros, M. J. Egenhofer, and E. Bertino, Eds. Berlin, Germany: Springer, 2005, pp. 291–307.

[8] G. Mishra, S. Mazumdar, and A. Pal, "Improved algorithms for the evacuation route planning problem," *J. Combinat. Optim.*, vol. 36, no. 1, pp. 280–306, 2018.

[9] S. Kim, B. George, and S. Shekhar, "Evacuation route planning: Scalable heuristics," in *Proc. 15th Annu. ACM Int. Symp. Adv. Geographic Inf. Syst.*, 2007, Art. no. 20.

[10] E. M. Cepolina, "Phased evacuation: An optimisation model which takes into account the capacity drop phenomenon in pedestrian flows," *Fire Saf. J.*, vol. 44, no. 4, pp. 532–544, 2009.

[11] P. Lin, S. M. Lo, H. C. Huang, and K. K. Yuen, "On the use of multi-stage time-varying quickest time approach for optimization of evacuation planning," *Fire Saf. J.*, vol. 43, no. 4, pp. 282–290, May 2008.

[12] J. Koo, "High-rise building evacuation problem with heterogeneous population including the disabled," Ph.D. dissertation, Dept. Ind. Manage. Eng., Pohang Univ. Sci. Technol., Pohang, South Korea, 2013.

[13] D.-J. Noh, J. Koo, and B.-I. Kim, "An efficient partially dedicated strategy for evacuation of a heterogeneous population," *Simul. Model. Pract. Theory*, vol. 62, pp. 157–165, Mar. 2016.

[14] J. J. Fruin, "Designing for pedestrians: A level-of-service concept," in *Highway Research Record*, no. 355. Washington, DC, USA: Pedestrians, 1971, pp. 1–15.

[15] J. J. Fruin, *Pedestrian Planning and Design*. New York, NY, USA: Metropolitan Association of Urban Designers and Environmental Planners, 1971.

[16] J. J. Jarvis and H. D. Ratliff, "Note—Some equivalent objectives for dynamic network flow problems," *Manage. Sci.*, vol. 28, no. 1, pp. 106–109, 1982.

[17] R. Fortet, "L'algebre de Boole et ses applications en recherche operationnelle," *Trabajos de Estadistica*, vol. 11, no. 2, pp. 111–118, 1960.

[18] F. Glover and E. Woolsey, "Technical note—Converting the 0-1 polynomial programming problem to a 0-1 linear program," *Oper. Res.*, vol. 22, no. 1, pp. 180–182, 1974.

[19] P. Hansen, "Methods of nonlinear 0-1 programming," *Ann. Discrete Math.*, vol. 5, pp. 53–70, Jan. 1979.

CHANG HYUP OH received the B.S. degree in industrial and management engineering from the Pohang University of Science and Technology (POSTECH), Pohang, South Korea, in 2014, where he is currently pursuing the Ph.D. degree in industrial and management engineering. His research interests include the development of mathematical models and heuristic algorithm for a building evacuation planning.

MIN HEE KIM received the B.S. degree in industrial and management engineering from the Pohang University of Science and Technology (POSTECH), Pohang, South Korea, in 2017. She is currently pursuing the Ph.D. degree with the Department of Industrial and Systems Engineering, University of Wisconsin–Madison, Madison, WI, USA. Her research interests are system degradation modeling, and prognostics and decision making with applications in healthcare and manufacturing.

BYUNG-IN KIM received the B.S. and M.S. degrees in industrial engineering from POSTECH, South Korea, in 1991 and 1994, respectively, and the Ph.D. degree in decision sciences and engineering systems from the Rensselaer Polytechnic Institute, Troy, NY, USA, in 2002. He was an Assistant Professor of industrial and systems engineering with the University of Memphis and the Director of the Research and Development with the Institute of Information Technology, Inc., The Woodlands, TX, USA. He is currently a Professor and the Head of the Department of Industrial and Management Engineering, POSTECH. His research interests include vehicle routing problems, industrial optimization problems, generic simulation, healthcare optimization, and logistics. He was a Senior Member of the IISE. He is a Member of INFORMS, KIIE, and KORMS. He received the Franz Edelman Finalist Award 2004 from INFORMS.

YOUNG MYOUNG KO received the B.S. and M.S. degrees in industrial engineering from Seoul National University, South Korea, and the Ph.D. degree in industrial engineering from Texas A&M University, USA. He is currently an Associate Professor of industrial and management engineering with the Pohang University of Science and Technology (POSTECH), South Korea and also serves as an Associate Editor of *IISE Transactions*. His research focuses on the analysis and optimization of large-scale stochastic systems such as service systems, telecommunication networks, ICT infrastructure, and renewable energy systems.