Contraflow Transportation Network Reconfiguration for Evacuation Route Planning

Sangho Kim, Shashi Shekhar, *Fellow*, *IEEE*, and Manki Min, *Member*, *IEEE*

Abstract—Given a transportation network having source nodes with evacuees and destination nodes, we want to find a contraflow network configuration (that is, ideal direction for each edge) to minimize the evacuation time. Contraflow lane reversal is considered a potential remedy to reduce congestion during evacuations in the context of homeland security and natural disasters (for example, hurricanes). This problem is computationally challenging because of the very large search space and the expense of calculating the evacuation time on a given network. To our knowledge, this paper presents the first macroscopic approaches for the solution of a contraflow network reconfiguration incorporating road capacity constraints, multiple sources, congestion, and scalability. We formally define the contraflow problem based on graph theory and provide a framework of computational structure to classify our approaches. A Greedy heuristic is designed to produce high-quality solutions with significant performance. A Bottleneck Relief heuristic is developed to deal with large numbers of evacuees. We evaluate the proposed approaches both analytically and experimentally using real-world data sets. Experimental results show that our contraflow approaches can reduce the evacuation time by 40 percent or more.

Index Terms—Contraflow, optimization, graph algorithms, heuristics design.

1 Introduction

Efficient evacuation route planning is currently an issue of major importance due to the increasing risks from both terrorist attacks and natural disasters. For transportation system planners, the main issue has been the severe traffic jams during the evacuation process. In the aftermath of Hurricanes Katrina and Rita in 2005, the transportation community observed the need for increased evacuation route capacity, as well as a more accurate estimate of the evacuation time [29]. Contraflow, or lane reversals, has been discussed as a potential remedy to solve such a tremendous congestion by increasing the outbound evacuation route capacity [41], [42]. Although contraflow is primarily important for evacuations, its applications are not limited to emergencies. Other examples of contraflow include the reversals of the two center lanes of the highway system in Washington, D.C., during morning and evening rush hours [7], [32] and road network reconfiguration after football games.

The contraflow problem for evacuation can be defined as follows: Given 1) a transportation network with edges, each having a capacity and a travel time, and 2) source and destination junctions, we find a reconfigured network identifying the ideal direction for each edge to minimize the evacuation time by reallocating the available capacity. Finding the optimal contraflow network configuration is considerably challenging because we may have to enumerate combinations of edge (that is, road segment) directions and compare those combinations by calculating the evacuation time. The task is NP-complete, and its proof is shown in this paper. In addition, it takes considerable time to evaluate each contraflow configuration candidate, taking the dynamics of traffic flow into account. Thus, we need to consider the balance between model realism and prohibitive computing requirements engendered by the exhaustive search space and the demands of the realistic modeling of traffic flow.

Hamza-Lup et al. [22] proposed algorithms to tackle the contraflow problem. Their approach is based on evacuation modeling with a single source, thus leading to finding the optimal paths to destinations and overlaying them. This planning approach does not take the overall capacity of the road network into account. Tuydes and Ziliaskopoulos designed a mesoscopic contraflow network model [39] based on a dynamic traffic assignment method. Their approach is subject to the problem of mathematical optimization, however, and thus, they have not shown scalable experiments. In addition, their Tabu-based heuristic approach [40] is a search-based iterative optimization technique. Although the Tabu search significantly reduces the search space to be explored, the search space may still be too large in cases of very large spatial networks due to the combinatorially increasing number of candidate networks reconfigured by contraflow.

To address the challenges of evacuation route planning, we introduce the parameter named Overload Degree, which classifies the computational structure of the contraflow reconfiguration problem by the ratio of the number of traveling units (for example, evacuees) to the bottleneck capacity (that is, minimum cut in graph theory) of the given network. We propose heuristics to determine the ideal

- *S. Kim is with the Geodatabase team, ESRI, 380 New York Street, Redlans, CA 92373. E-mail: skim@esri.com.*
- *S. Shekhar is with the University of Minnesota, 4-192, EE/CS Bldg., 200 Union Street SE, Minneapolis, MN 55455. E-mail: shekhar@cs.umn.edu.*
- *M. Min is with the Electrical Engineering and Computer Science Department, South Dakota University, 133 Administration Bldg., Brookings, SD 57007. E-mail: manki.min@sdstate.edu.*

Manuscript received 1 May 2006; revised 2 Feb. 2007; accepted 23 Oct. 2007; published online 6 Nov. 2007.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDESI-0220-0406.
Digital Object Identifier no. 10.1109/TKDE.2007.190722.

1041-4347/08/$25.00 © 2008 IEEE Published by the IEEE Computer Society

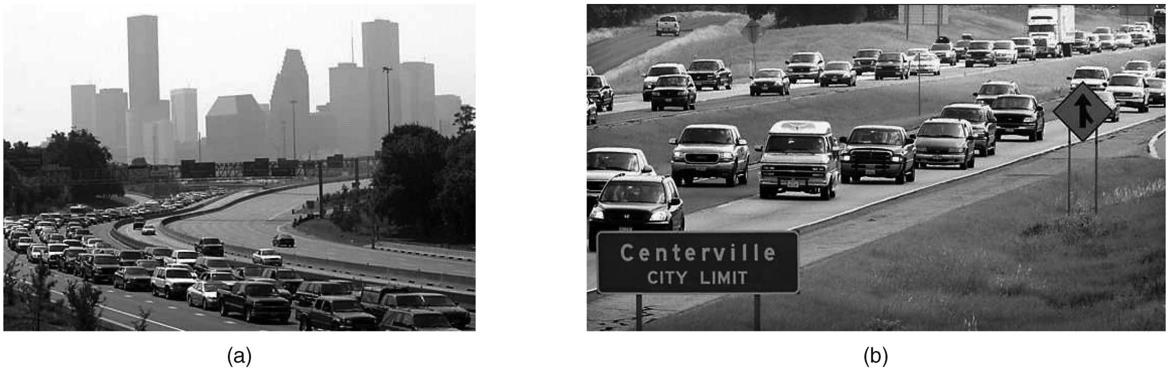


Fig. 1. Before contraflow, the northbound lanes of Interstate Highway 45 in Houston were jammed during the Hurricane Rita evacuation. After contraflow, the northbound traffic is smoother on both sides of Interstate Highway 45. (Source: wfaa.com). (a) Before contraflow. (b) After contraflow.

direction of edges in transportation networks for evacuation. The Greedy heuristic runs an evacuation route planner to determine the condition of congestion on a given original configuration and flips highly congested road segments in a greedy manner. The Bottleneck Relief heuristic identifies the bottleneck of a given network and increases the bottleneck capacity by contraflow.

We evaluated our approaches using analytical and experimental validation methodologies. In the experimental evaluation, we prepared real-world data sets to test the performance and scalability of the approaches. Experimental results and case studies show that the proposed approaches can reduce the evacuation time by 40 percent or more. In addition, we present findings with important implications for planners and first responders as they prepare contraflow evacuation schemes.

1.1 Motivation

Evacuation route planning has become a topic of critical importance due to the September 11 terrorist attacks and recent catastrophic hurricanes that required large-scale evacuations across the US. In 2005, two major hurricanes, Katrina and Rita, hit the southeastern part of the US and caused severe damage across several coastal states [36]. Especially during the Rita evacuation, a greater number of evacuees than expected followed the evacuation order with their personal vehicles. The following are quoted observations [29] of the traffic problems that occurred during the Rita evacuation:

- **Congestion problem.** "An estimated three million people evacuated the Texas coast, creating colossal, 100-mile-long traffic jams that left many stranded and out of fuel. Drivers heeding the call to evacuate Galveston Island and other low-lying areas took four to five hours to cover the 50 miles to Houston, and from there roadway conditions were even worse, with traffic crawling at just a few miles per hour.... After crawling only 10 or 20 miles in nine hours, some drivers turned around to take their chances at home rather than risk being caught in the open when the hurricane struck."
- **Contraflow problem.** "High-occupancy vehicle (HOV) lanes went unused, as did many inbound lanes of highways, because authorities inexplicably waited until late Thursday to open some up.... As congestion worsened, state officials announced that contraflow lanes would be established on Interstate Highway 45 (Fig. 1b), US Highway 290, and Interstate Highway 10. But by midafternoon, with traffic immobile on US Highway 290, the plan was dropped, stranding many and prompting others to reverse course. "We need that route so resources can still get into the city," explained an agency spokeswoman."

During the Rita evacuation, transportation analysts [29] were able to observe the inefficient use of road capacity and the effects from the ill-planned contraflow, which resulted in disorganized movement of people. They listed the failure to use contraflow lanes and road shoulders for evacuation traffic as one of the planning problem lessons learned from Katrina and Rita.

Although it is a subject of recent dramatic interest, contraflow has other routine but important applications. One application is the use of reversible lanes to deal with morning and evening peak commuter time. Washington State has been operating reversible two-lane roadways for peak period HOV-3 vehicles [7], [32]. The reversible lane system has been reported to provide significant savings in travel time. A second application of contraflow is common during special events when all lanes are reversed to accommodate outbound traffic at the end of a sporting event or concert. This is a special case of contraflow, having a single source with multiple destinations.

1.2 Related Work and Our Contribution

The material and literature on evacuations in general and the contraflow problem in particular have been published in various domains including social and behavioral sciences, transportation, and mathematics [11], [12]. A survey [42] of evacuation issues and contraflow revealed that transportation planners have no recognized standards or guidelines for the design, operation, and location of contraflow segments. Many states threatened by hurricanes and considering contraflow plans are dependent on past evacuation experiences. Litman [29] identified the planning problems of Hurricanes Katrina and Rita and specifically criticized unplanned contraflow orders and the failure to use contraflow lanes.

Past papers and Department of Transportation reports [15], [37], [41], [42] have mainly tackled the managerial and operational aspects of contraflow, such as signal control, merging, and cost. When planners design a network configuration for evacuation scenarios, they mainly depend on empirical guesses. Such handcrafted contraflow plans have revealed that they are neither efficient for finding critical road segments of contraflow nor flexible for accommodating various variables [13], [14].

Hamza-Lup et al. [22] introduced two different contraflow algorithms from a computer science perspective: one based on a multicast routing problem and the other based on breadth-first graph traversal. These algorithms can handle only a single coordinated incident due to the conflicts of multiple optimal paths that occur in multiple-source and multiple-destination evacuation models. The authors did not clearly describe the use of different link capacities. Thus, their approach is not effective when the number of traveling units is finite, road capacities are constrained, specific destination nodes are prescribed, or evacuees are spread over many locations.

Tuydes and Ziliaskopoulos [39] proposed a mesoscopic contraflow network model based on a dynamic traffic assignment method. They formulated capacity reversibility using a mathematical programming method. The discretized hypothetical network required to solve the traffic assignment problem, however, hindered large-scale network scenarios from running in their framework. They also tried a Tabu-based heuristic approach [40] to address capacity reversibility optimization. Their solutions required a considerable number of iterations, thus limiting their input to a small network.

Theodoulou and Wolshon [38] used CORSIM microscopic traffic simulation to model the freeway contraflow evacuation around New Orleans. With the help of the microscale traffic simulator, they were able to suggest alternative contraflow configurations at a detailed level. However, the microscopic simulation model requires labor-intensive network coding and significant runtime for each scenario, making it difficult to take advantage of spatial databases or easily compare alternative configurations. Evacuation route planning with other microscopic traffic simulation (for example, MITSIMLab [25]) has shown similar limitations.

Our contributions. Previously, Kim and Shekhar [28] proposed two heuristics for contraflow planning. One heuristic, named Flip High Flow Edge (FHFE), is based on a greedy algorithm with a flow history of edges. The FHFE generates a suboptimal contraflow plan without iteration. The other heuristic is based on a simulated annealing optimization technique. Due to the searching property (that is, global optimization) of simulated annealing, it can generate slightly better solutions than FHFE, although the gain from the simulated annealing method is relatively small despite its long runtime by iterative search.

In this paper, we present capacity-aware global contraflow heuristics that are designed to handle multiple source and destination nodes. We classify the contraflow problem using Overload Degree and present two heuristics. Our Greedy heuristic is designed to handle scenarios with a significantly large population and network size. It also has a flexible algorithm structure by using an evacuation route planner as a plug-in module, thus leaving room for improvements with faster evacuation route planners. The other proposed heuristic is a Bottleneck Relief heuristic, which tackles the inherent congestion problem of contraflow by identifying bottlenecks in the network using a minimum cut.

Analytical and experimental evaluations are provided to validate the usefulness of the proposed approaches. Experimental results show that less than 30 percent of the total edges for contraflow is enough to reduce evacuation time by more than 40 percent. We also provide a comparison of solution quality between the proposed heuristics and integer programming (IP) (optimal contraflow network generator).

1.3 Scope and Outline of the Paper

Our evacuation modeling is based on graph theory with flow analysis on a macroscopic flow model. Our modeling does not include the social behavior of the evacuees, the operational cost/policy of contraflow, or traffic signals. Our focus is to design scalable contraflow heuristics to address large-scale transportation networks and accurately compare the performance between a given network and a contraflow-reconfigured network within our computational framework.

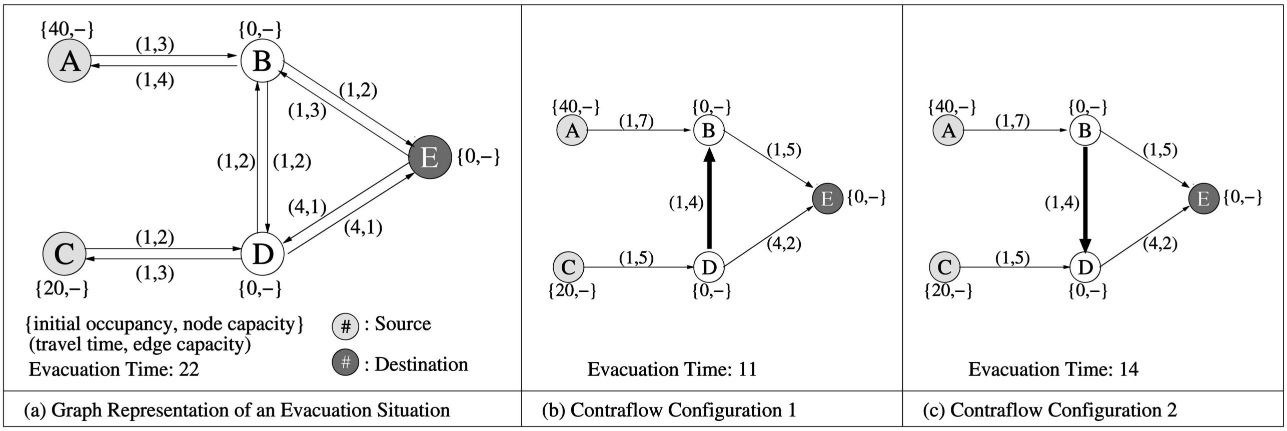
The rest of the paper is organized as follows: Section 2 presents the modeling and hardness of the evacuation problem. Section 3 provides a computational framework of the contraflow problem and presents our proposed approaches to solving the contraflow problem. In Section 4, we describe design decisions and present their analytical evaluations. Section 5 gives the experimental setup and evaluation of the approaches. Finally, Section 6 summarizes and concludes with a discussion of future work.

2 Modeling and Problem Hardness

2.1 Problem Formulation

Evacuation is a situation where residents in a dangerous area are removed to safe places as quickly as possible. Many approaches have been proposed to model the evacuation situation using microscopic simulation [8], [3], [31], [34], mesoscopic models [39], and macroscopic network flow models [2], [10], [21]. Microscopic simulation models traffic flow at a single-vehicle level. The behavior of individual drivers is under the influence of vehicles in their proximity. This model is usually accompanied by car-following models. Mesoscopic simulation models traffic flow by groups of traffic entities. Vehicles are not described individually but in more aggregate terms using probability. Macroscopic models describe traffic flow at a high level of aggregation. Although researchers have debated the suitability of these various approaches for describing traffic flow, many favor macroscopic modeling due to the increased public attention, improved techniques, and the computational capacity this approach offers [24].

In this research, we select a macroscopic network flow model using a mathematical graph to represent the evacuation situation. The movement of evacuees is represented as a flow on the graph. Although a macroscopic model loses the properties of single-vehicle traffic flow (for example, congestion propagation), it is still a powerful tool for evacuation planning because it effectively deals with the road density, the weighted mean speed, and, most importantly, the capacity of a given transportation network.



(a) Graph Representation of an Evacuation Situation

(b) Contraflow Configuration 1

(c) Contraflow Configuration 2

Fig. 2. Graph representation of a simple evacuation situation and two following contraflow configuration candidates. (a) Graph representation of an evacuation situation. (b) Contraflow configuration 1. (c) Contraflow configuration 2.

Thus, macroscopic models provide evacuation planners with the means to evaluate systemwide evacuation strategies. In addition, macroscopic models are more suitable than microscopic models for large-size networks due to their scalability.

In macroscopic evacuation models, it is necessary to represent the situation with a mathematical graph structure. Let G(N, E) be a directed network, where N is the set of nodes, and E is the set of edges. Each node has an initial occupancy value—that is, the number of residents to evacuate—and a node capacity. The use of node capacity depends on the evacuation model. For example, a building evacuation may model a room as a node. In this case, room size is a node capacity. If a node is modeled as an intersection in a transportation network, the node capacity can be set to infinity. In the case of limited node capacity, such as a small stairwell in a building evacuation or a toll plaza in a transportation network, such a node will be a choking junction and negatively affect the edge flow around the node.

Each edge also has an edge capacity, a travel time, and an initial direction. The edge capacity is defined as the number of traveling units (for example, vehicles or pedestrians) per given unit period. For example, a highway edge segment may have a capacity of 1,800 vehicles per hour under a normal operation. The evacuation situation has multiple source nodes and destination nodes. Evacuation time is defined as the period from the moment when the first evacuee leaves a source node to the moment when the last evacuee arrives at a destination node.

It is worthwhile to note that there are two different ways of modeling edge capacity in macroscopic methodology. The first method can be called "continuous entering." In this approach, it is assumed that the evacuees equivalent to the edge capacity keep entering an edge every unit of time as long as the edge is available. The second method can be called "occupy and empty." Evacuees equivalent to the edge capacity occupy the edge for the edge travel time. During the occupying period, the edge is not available to other evacuees staying at a "from" node. In our evacuation model, we choose "continuous entering" to model the edge capacity because it is a more realistic representation of the evacuees' movement.

With the given network setup, we want to find a network reconfigured by contraflow with the objective of minimizing the evacuation time. There are two constraints. First, the capacity needs to be constant. Our problem formulation does not incorporate the dynamic nature of network properties (for example, bridge collapse in the middle of an evacuation procedure). On the other hand, the travel time of an edge can be either constant or changeable, depending on the characteristics of the evacuation route planner. Second, partial reversal is not allowed. This constraint will keep the problem size at a reasonable level. In our modeling, we follow a typical network model, which is easily persisted in relational tables, taking future enhancements into account [23]. The following is a formal summary of our contraflow problem:

Given: 1. A transportation network, a directed graph G(N, E).
2. Each node has initial occupancy and capacity.
3. Each directed edge has a capacity, a travel time, and an initial direction.
4. Source and destination nodes.

Find: A contraflow network configuration (that is, desired direction for each edge).

Objective: Minimize evacuation time.

Constraint: 1. Capacity is constant.
2. Partial reversal (for example, partial number of lanes) is not allowed.

Fig. 2 illustrates a simple evacuation situation based on the problem formulation. Nodes A and C are modeled as source nodes, whereas node E is a destination node (for example, evacuation shelter). Nodes B and D have no initial occupancy and only serve as transshipment nodes. The unit time (for example, minute) is defined according to the model scale. The evacuation time of the original network in Fig. 2a is 22 (details of how to measure evacuation time are discussed in Section 4.2). Figs. 2b and 2c illustrate two possible contraflow configurations. All the two-way edges used in the original configuration are merged by capacity and directed in favor of increasing the outbound evacuation capacity. There are two candidate configurations that differ in the direction of edges between nodes B and D. The network in Fig. 2b reduces the evacuation time to 11 (that is, 50 percent of the original evacuation time), whereas the network in Fig. 2c reduces evacuation time to 14 (64 percent). This example illustrates the importance of choice among possible contraflow network configurations. Moreover, we have to know that there are critical edges affecting the evacuation time such as edge (B-D) in Fig. 2.

2.2 Proof of NP-Completeness

In this section, we prove the NP-completeness of the contraflow problem. In general, the process of devising an NP-completeness proof for a decision problem Π consists of the following four steps [17]:

1. *showing that* Π *is in NP,*
2. *selecting a known NP-complete problem* Π′,
3. *constructing a transformation* f *from* Π′ *to* Π, and
4. *proving that* f *is a (polynomial) transformation.*

To conduct our proof, we select the 3-SATISFIABILITY (3SAT) problem as our known NP-complete problem. This problem is considered the root of most other NP-complete problems and is derived from the SATISFIABILITY problem whose NP-completeness was proven by Cook [17]. The 3SAT problem is specified as follows:

3SAT
INSTANCE: Collection $\mathrm{C} = c_1, c_2, \ldots, c_m$ of clauses on a finite set U of variables such that $|c_i| = 3$ for $1 \leq \mathrm{i} \leq \mathrm{m}$.
QUESTION: Is there a truth assignment for U that satisfies all the clauses in C?

The *EVAC-TIME* used in the following definition is a polynomial function that can calculate the evacuation time of a given graph. For simplicity, each edge in an undirected graph G could be flipped in either direction.

CONTRAFLOW
INSTANCE: An undirected graph $\mathrm{G} = (\mathrm{V}, \mathrm{E})$ with initial occupancy $o(\mathrm{v}) \in \mathrm{Z}^+$ (where Z^+ denotes the positive integers) for some $\mathrm{v} \in \mathrm{V}$, destination vertices for some $\mathrm{v} \in \mathrm{V}$, capacity $c(\mathrm{e}) \in \mathrm{Z}^+$ and travel time $t(\mathrm{e}) \in \mathrm{Z}^+$ for each $\mathrm{e} \in \mathrm{E}$, a directed graph $\mathrm{G}' = (\mathrm{V}, \mathrm{E}')$, and evacuation time bound $\mathrm{B} \in \mathrm{Z}^+$.
QUESTION: Is there a function $f : \mathrm{e} \rightarrow [\{\mathrm{u}, \mathrm{v}\}, \{\mathrm{v}, \mathrm{u}\}]$ for each $\mathrm{e} \in \mathrm{E}$ where $\{\mathrm{u}, \mathrm{v}\}$ or $\{\mathrm{v}, \mathrm{u}\}$ is a directed edge in E' such that $EVAC - TIME(\mathrm{G}') \leq \mathrm{B}$?

Lemma 1. *CONTRAFLOW is NP-complete.*

Proof. It is easy to see that CONTRAFLOW ∈ NP, since a nondeterministic algorithm need only guess a new directed graph G' by flipping all edges randomly and check in polynomial time that G' has evacuation time B or less.

We transform 3SAT to CONTRAFLOW. Let $\mathrm{U} = \{\mathrm{u}_1, \mathrm{u}_2, \ldots, \mathrm{u}_n\}$ and $\mathrm{C} = \{\mathrm{c}_1, \mathrm{c}_2, \ldots, \mathrm{c}_m\}$ be any instance of 3SAT. We must construct a graph $\mathrm{G}' = (\mathrm{V}, \mathrm{E}')$ and set a positive integer B such that G' has evacuation time B or less if and only if C is satisfiable.

The construction consists of a source component, a destination component, and a flipping component between the source and destination components. The source component consists of vertices $\mathrm{s}_1, \mathrm{s}_2, \ldots, \mathrm{s}_m$ with $o(\mathrm{s}) = 1$. The destination component consists of two layers. The first layer consists of each literal and their negated literals in U (that is, $\mathrm{u}_1, \overline{u}_1, \mathrm{u}_2, \overline{u}_2, \ldots, \mathrm{u}_n, \overline{u}_n$). The second layer consists of the XOR of each pair of literals (that is, $\mathrm{u}_1 \oplus \overline{u}_1, \mathrm{u}_2 \oplus \overline{u}_2, \ldots, \mathrm{u}_n \oplus \overline{u}_n$). This XOR layer serves as a destination node set in the CONTRAFLOW problem. The two nodes in a pair (u_i and $\overline{u}_i$) in the first layer are connected to each XOR node ($\mathrm{u}_i \oplus \overline{u}_i$) in the second layer with edges, each of which has $t(\mathrm{e}) = 1$ and $c(\mathrm{e}) = 1$. Finally, a flipping component consists of edges with the following definition: For each clause $\mathrm{c}_j \in \mathrm{C}$, let the three literals in c_j be denoted by x_j, y_j, and z_j. Then, the edges are $\{\mathrm{s}_j, \mathrm{x}_j\}$, $\{\mathrm{s}_j, \mathrm{y}_j\}$, $\{\mathrm{s}_j, \mathrm{z}_j\}$, each of which has $t(\mathrm{e}) = 0$ and $c(\mathrm{e}) = 1$. Fig. 3 shows an example of the contraflow graph obtained when $\mathrm{U} = \{\mathrm{u}_1, \mathrm{u}_2, \mathrm{u}_3, \mathrm{u}_4\}$ and $\mathrm{C} = \{\{\mathrm{u}_1, \overline{u}_3, \overline{u}_4\}, \{\overline{u}_1, \mathrm{u}_2, \overline{u}_4\}\}$.

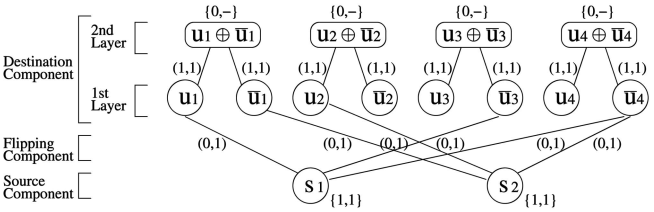


Fig. 3. CONTRAFLOW instance resulting from 3SAT instance in which $\mathrm{U} = \{\mathrm{u}_1, \mathrm{u}_2, \mathrm{u}_3, \mathrm{u}_4\}$ and $\mathrm{C} = \{\{\mathrm{u}_1, \overline{u}_3, \overline{u}_4\}, \{\overline{u}_1, \mathrm{u}_2, \overline{u}_4\}\}$.

It is easy to see how the construction can be accomplished in polynomial time. All that remains to be shown is that C is satisfiable if and only if $EVAC - TIME(\mathrm{G}') \leq \mathrm{B}$ by flipping edges in G to prove that the above construction is indeed a transformation.

→ : Suppose that C is satisfiable. We define the function f as $\mathrm{e} = \{\mathrm{u}, \mathrm{v}\}$ if v is TRUE or $\mathrm{e} = \{\mathrm{v}, \mathrm{u}\}$ if v is FALSE (that is, draw an arrow head on the TRUE node and an arrow tail on the FALSE node). We assume that B is equal to the number of source nodes. If C is satisfiable, at least one edge from each source node will be directed toward the destination component. This guarantees that one occupant in each source node can evacuate to the destination nodes (second layer in the destination component) with at most B evacuation time. The worst-case evacuation time B happens when all the source nodes are pointed to one node in the first layer of the destination component.

← : Suppose that $EVAC - TIME(\mathrm{G}') \leq \mathrm{B}$ by using the same flipping function f described above. For each occupant in each source node to evacuate to a destination node, at least one edge from the source node should be directed toward the first layer of the destination component. This guarantees that C is satisfiable. □

3 Computational Framework and Proposed Approaches

3.1 Computational Framework and Approach Overview

In this section, we introduce the computational structure of the contraflow problem using Overload Degree and present appropriate approaches according to each workload zone.

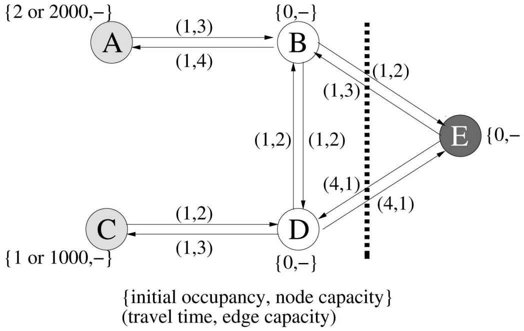


Fig. 4. Simple evacuation network with bottleneck capacity 3.

As will be shown, Overload Degree is a key determinant of the overall evacuation time and need for contraflow.

Overload degree. Due to the combinatorial nature of the contraflow problem, acquiring the optimal solution becomes considerably challenging as the size of the network increases. To address the challenges in problem size, we need to define a parameter to classify the computational structure of the problem. The number of traveling units and bottleneck capacity of a given network are two critical factors affecting the computational structure. The evacuees' movement is analogous to the flow movement through a bottleneck of a gourd bottle. If the amount of flow is large or bottleneck size is small, it takes a long time to finish the flow movement. With this analogy, we define the term "Overload Degree" as follows:

Definition 1 (Overload Degree). *Overload Degree = Number of Traveling Units/Bottleneck Capacity Without Contraflow.*

"Bottleneck Capacity Without Contraflow" refers to a minimum cut value (or maximum flow value) of a given network without contraflow. In the calculation of the bottleneck capacity, node capacity is not considered. The hardness of the contraflow problem is a function of the Overload Degree. For a small Overload Degree, we can consider mathematical programming, search-based approaches, or microscopic simulation to produce optimal results. For example, suppose that there are 500 evacuees on a network whose minimum cut is 100. The Overload Degree is only 5. In such a case, the network has enough bottleneck capacity to evacuate the 500 evacuees. Thus, the computational workload is relatively low. For a medium Overload Degree, we definitely need heuristic approaches to achieve a balance between the result quality and the reasonable computational workload. For a large Overload Degree, we need a more computationally efficient approach.

Fig. 4 is the same example evacuation network as in Fig. 2 but with different initial occupancy. The dotted line is a bottleneck of this network, separating the source nodes and the destination nodes. The value of the bottleneck capacity is found to be 3 by adding the capacity of edge (B-E) (that is, from B to E) and (D-E). Suppose node A has occupancy 2 and C has 1. We do not need contraflow because the current bottleneck capacity is enough to handle the small number of traveling units (that is, evacuees). In this case, the Overload Degree is 1. As the number of initial occupancy increases (for example, > 3), the current bottleneck capacity becomes insufficient. We start thinking of contraflow to reduce the evacuation time. The computational workload accordingly becomes heavy to calculate the scheduling of the large number of traveling units. Suppose nodes A and C have 2,000 and 1,000 evacuees each. Then, the situation becomes close to an infinite source problem as we can ignore the transitional starting and ending periods of evacuation. As shown in this example, Overload Degree is a critical parameter for determining the problem size and its congruent solution for a given network.

In the absence of overload (for example, Overload Degree is less than 1), contraflow offers few or no benefits because the original network has enough capacity for the current evacuees to pass through. If the Overload Degree is small (for example, a one-digit number), it is computationally feasible to identify "optimal" contraflow configurations by using optimization techniques such as mathematical programming, search-based optimization, or microscopic simulation. Our IP formulation belongs to the small Overload Degree category. Results from the IP formulation are useful to assess the quality of solutions obtained by our heuristics.

If the Overload Degree is medium, we have to consider heuristics due to the heavy computational workload. At this level of workload, it is impossible to use an iterative learning process, which is only feasible in a small Overload Degree. We suggest a noniterative heuristic based on a greedy approach.

Last, if the Overload Degree is large, it is close to the case where the network has an infinite source of evacuees. Here, it is necessary to simplify the evacuation modeling to address such a heavy computational workload. We have designed a minimum cut and maximum flow [16] based Bottleneck Relief heuristic that ignores the amount of the population constraint.

Use of an evacuation route planner. The role of an evacuation route planner in our framework is to calculate the flow history and evacuation time of a given network. The flow history of an edge is equivalent to the total number of traveling units that pass through the edge during an evacuation time. Our contraflow solution framework separates the evacuation route planner from the contraflow reconfiguration algorithm. Thus, the evacuation route planner to be plugged in can take either a microscopic or a macroscopic simulation approach as long as the planner conforms to the input/output rules.

Fig. 5 shows how the evacuation route planners function within our proposed framework. The input to the system is an original evacuation network with predefined source/destination nodes and edges with capacity and travel time. There are three algorithmic components: IP, a Greedy heuristic, and a Bottleneck Relief heuristic. For the IP approach, the evacuation route planner is combined with a mathematical optimizer to evaluate the networks generated by iterative enumeration and serve as an objective function. The Greedy heuristic uses the flow history of the original network as input and produces a reconfigured contraflow network. The Bottleneck Relief heuristic uses the original network as input and directly produces a reconfigured contraflow network.

The cost of running an evacuation route planner increases with the size of the network. Thus, how the evacuation route planner is used is critical in the framework. The Greedy heuristic uses the evacuation route

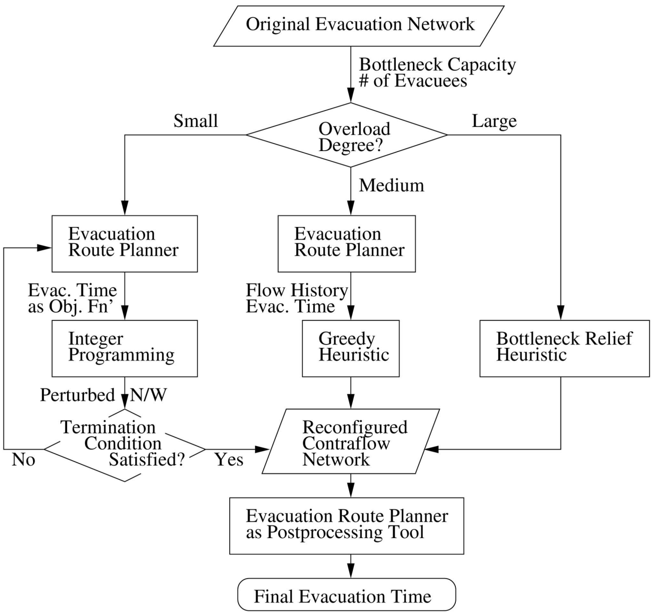


Fig. 5. Use of an evacuation route planner in the contraflow solution framework.

planner once to generate a contraflow network. The Bottleneck Relief heuristic does not use an evacuation route planner. On the other hand, the IP approach uses the evacuation route planner iteratively.

3.2 Greedy Heuristic

The basic assumption of the Greedy heuristic is that when we run an evacuation route planner over an original network configuration without contraflow, the edges having more congestion history are more influential in the decision of edge flips. Therefore, it is necessary to quantify the congestion history on each edge with the data from the evacuation route planner. We define the $FlowHistory$ and $CongestionIndex$ of an edge e in the following way:

Definition 2 (flow history). $FlowHistory(e)$ = *Total number of traveling units going through edge* e *during* $EvacuationTime$.

Definition 3 (congestion index). $CongestionIndex(e) = FlowHistory(e)/(Capacity(e) \times EvacuationTime)$.

$FlowHistory(e)$ is acquired from the result of the evacuation route planner. The denominator in Definition 3 refers to the maximally possible amount of flow of edge e during $EvacuationTime$. Thus, $CongestionIndex(e)$ indicates the percentage of edge utilization during $EvacuationTime$. A higher $CongestionIndex(e)$ value means that the edge e has been more congested during the evacuation process.

The third definition used in the greedy approach is the "Degree of Contraflow." We can define the Degree of Contraflow in the reconfigured network as follows:

Definition 4 (Degree of Contraflow). *Degree of Contraflow(DoC) = Number of Flipped Edges/Total Number of Edges.*

This percentage parameter indicates how many edges are flipped among all edges in the reconfigured network. Our Greedy heuristic has the ability to control this parameter, which is important in the context of evacuation because unnecessary flips lead to the waste of resources. That is, more emergency professionals are needed as the number of reversed road segments increases. In addition, the unflipped edges (that is, in-bound road segments) can be used as capacity for incoming emergency vehicles (for example, ambulances and fire trucks).

Algorithm 1. Greedy

```
run an evacuation route planner to produce
  $FlowHistory$ and $EvacuationTime$ on $G_{original}$;
for all edge $e \in G_{original}$ do
    $CongestionIndex(e) = FlowHistory(e)/(Capacity(e)$
      $\times EvacuationTime)$;
end for
sort edges by $CongestionIndex(e)$ in descending order;
$G_{reconfigured} = G_{original}$;
for all $(i, j)$ in the first $DoC\%$ edges in the sorted edges
do
    $G_{reconfigured}.flip((i, j))$;
end for
return $G_{reconfigured}$;
```

The Greedy algorithm shown in Algorithm 1 works in the following way: First, we run any evacuation route planner to generate the flow history and evacuation time of a given original network. Second, we assign a congestion index value to each edge. Third, the edges are sorted by congestion index in descending order. Finally, we flip edges in favor of a higher congestion index value among the first $DoC\%$ of the sorted edge set. The evacuation route planner must be run again over the reconfigured network to get the evacuation time of the reconfigured network.

This noniterative algorithm structure may result in a network disconnection problem because the algorithm suggests a reversal that disconnects two subnetworks. We can show that such a disconnection problem does not happen. Suppose that we have two networks, G_1 and G_2, and they are connected by two bidirectional edges, e_1 (from G_1 to G_2) and e_2 (from G_2 to G_1). G_1 has S_1 (source) and D_1 (destination). G_2 has S_2 (source). Assume to the contrary that G_2 is disconnected by reversing the edge e_2. The disconnected network means that there is at least one route from G_1 to G_2 generated by the evacuation route planner. Such a route does not exist because G_2 does not have any destination, contradicting our assumption.

Example. Fig. 6 shows a series of steps using the Greedy algorithm to generate a contraflow network from a given original network. We assume that the given Degree of Contraflow is 60 percent. The network in step 1 is the given original network. If we run an evacuation route planner on the network, we acquire the flow history, as well as the evacuation time. An optimal route planner produces evacuation time 22. The network in step 2 shows the flow history value. For example, the value 17 over edge (D-B) means that 17 evacuees pass through the edge during evacuation time 22. In step 3, the congestion index values are generated from the information of step 2 using the formulation in the congestion index definition. The congestion index values are sorted

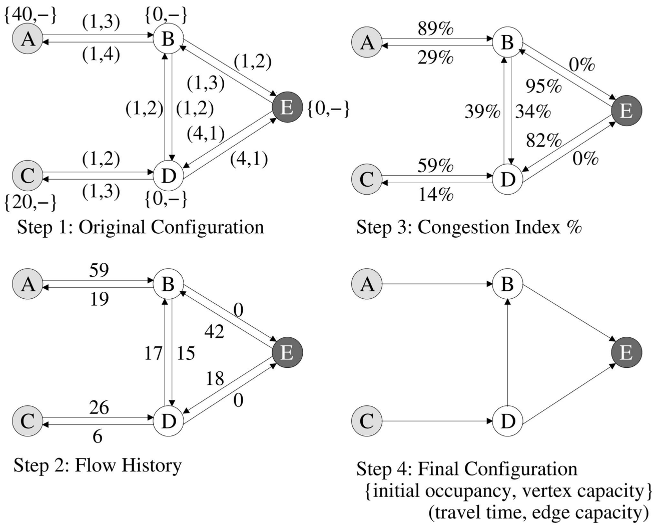


Fig. 6. Example of the Greedy algorithm.

in descending order, and the first 60 percent of them (underlined edges) are greedily selected, as shown in Table 1. Each selected edge is compared with its opposite edge, and the opposite edge is flipped if the selected edge wins. The final reconfigured network is shown in step 4, after the flipping process is finished.

The flow on edge (B-A) or (D-C) can be generated in step 2 because some amount of flow oscillates between the two nodes. This may not happen in an actual evacuation scenario but may happen in a flowgraph. The oscillation does not affect the final evacuation time. The final decision between nodes B and D is edge (D-B) because the direction from D to B shows more congestion, as seen in step 3 in Fig. 6.

3.3 Bottleneck Relief Heuristic for a Large Overload Degree

The Bottleneck Relief approach starts from the well-known theorem by Ford and Fulkerson [16], which states that "The value of the max-flow in a capacitated network is equal to the value of the min-cut." In the context of transportation networks, the min-cut is a bottleneck or choke capacity. The idea behind this approach is to identify the bottleneck and increase its capacity by contraflow.

Algorithm 2. BottleneckRelief

```
while $maxflow_{new} > maxflow_{old}$ do
    find $mincut$ of $G$;
    flip edges across $mincut$ toward destination;
    $maxflow_{old} = maxflow_{new}$;
    $maxflow_{new} = \mathrm{maxflow}(G)$;
end while
return $G$;
```

TABLE 1
Sorted Congestion Indexes from Step 3 in Fig. 6

Edge	CI (%)	Edge	CI (%)
B-E	95	B-D	34
A-B	89	B-A	29
D-E	82	D-C	14
C-D	59	E-B	0
D-B	39	E-D	0

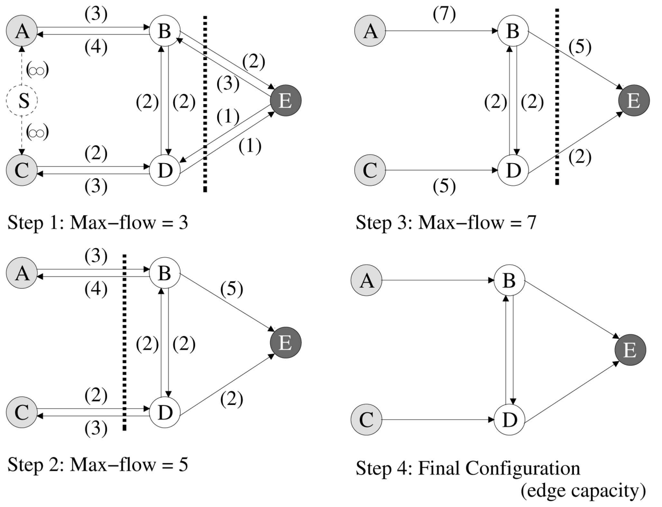


Fig. 7. Example of the Bottleneck Relief heuristic.

If the given graph G has multiple sources and multiple destinations, we have to place a supersource connecting to the sources with infinite capacity and a superdestination connecting to the destination with infinite capacity before the algorithm BottleneckRelief is applied. The algorithm BottleneckRelief, shown in Algorithm 2, finds a min-cut of the given graph and flips edges across the min-cut. Then, the location of the min-cut will change. The algorithm keeps finding the min-cut until the maximum flow does not improve. This algorithm is suitable for a network having a large Overload Degree because the maximum flow is based on the infinite flow from sources to sinks. Evacuation scenarios over heavily crowded areas and reversible highway systems for specified periods of time are examples to which we can apply this algorithm. Suppose that the original network has p number of occupancy, n vertices, and m edges. A proposed randomized algorithm [26] finds a minimum cut with high probability in $O(m \log^3 n)$. In the worst case, our Bottleneck Relief heuristic runs m times, which leads to $O(m^2 \log^3 n)$ runtime. The network disconnection problem does not happen by the Bottleneck Relief heuristic. The proof is the same as for the Greedy heuristic.

Example. Fig. 7 illustrates the application of the Bottleneck Relief heuristic to our simple graph. Nodes A and C are still source nodes, whereas node E is a destination node. The source nodes are connected from a supersource as shown in step 1. The min-cut (or max-flow) in the original graph is represented as a dotted line in step 1 and has value 3. In step 2, we flip edges across the first min-cut in favor of increasing capacity to the destination. Then, the previous min-cut is no longer a min-cut due to its increased capacity by contraflow. A second min-cut is also shown as a dotted line in step 2. We continue these steps until the max-flow does not increase. Step 4 shows the final network reconfiguration.

TABLE 2
Contraflow IP Formulation

#	Equation	Description
(1)	$\min \sum_{t=1}^{T} F_t$	Objective function (i.e., minimize evacuation time)
(2)	$n_{i,t} = n_{i,t-1} + \sum_{(j,i)\in E} f_{(j,i),t-ET_{(j,i)}} + \sum_{(i,j)\in E} g_{(j,i),t-ET_{(i,j)}}$ $- \sum_{(i,j)\in E} f_{(i,j),t} - \sum_{(j,i)\in E} g_{(i,j),t},$ $\forall\ i \in N, t = 1, \ldots, T$	Flow conservation constraint
(3)	$\begin{cases} cf_{(i,j)} + cf_{(j,i)} \leq 1, & \forall i,j \text{ s.t. } (i,j),(j,i) \in E \\ cf_{(i,j)} \leq 1, & \forall i,j \text{ s.t. } (i,j) \in E, (j,i) \notin E \\ cf_{(j,i)} \leq 1, & \forall i,j \text{ s.t. } (j,i) \in E, (i,j) \notin E \end{cases}$	Contraflow flipping constraint
(4)	$\begin{cases} f_{(i,j),t} \leq EC_{(i,j)}(1 - cf_{(i,j)}), & \forall\ (i,j) \in E, t = 0, \ldots, T \\ g_{(i,j),t} \leq EC_{(j,i)} cf_{(j,i)}, & \forall\ (j,i) \in E, t = 0, \ldots, T \end{cases}$	Contraflow capacity constraint
	F_t: 1 iff there is flow on any edge at interval $(t-1, t]$	$cf_{(i,j)}$: 1 iff edge (i,j) is used for contraflow
	$n_{i,t}$: number of occupancy at node i during interval $(t, t+1]$	T: predetermined upper bound of total travel time
	$f_{(i,j),t}$: amount of flow on edge (i,j) during interval $(t, t+ET_{(i,j)}]$	$g_{(i,j),t}$: amount of flow on edge (j,i)
	$ET_{(i,j)}$: edge travel time, $EC_{(i,j)}$: edge capacity	N: set of nodes, E: set of edges

3.4 Integer Programming Formulation

The IP approach produces an optimal contraflow plan that can minimize the evacuation time. The IP approach is useful in comparing the solution quality between the proposed heuristics and optimal plan. Due to limited space, we introduce only the most important formulations used in the IP experiment.

Table 2 shows selected formulations of the IP approach. Equation (1) in Table 2 defines the objective function such that $\sum_{t=1}^{T} F_t$ is the total amount of time to finish the evacuation, assuming that T is large enough. If T is set to less than the minimum value, then the formulation becomes infeasible. Equation (2) describes the flow conservation constraints, meaning that inflow equals outflow. Equation (3) is a contraflow constraint, and it restricts the selection of contraflow as follows: When there is only one edge between two nodes, we have only two options: normal flow or contraflow. When there are two edges between two nodes, we have three options: two normal flows or one contraflow. We do not consider the case of reversing the two edges at the same time. Equation (4) is used to ensure the proper allowed amount of flow on each edge based on the value of the edge capacity.

4 Design Decisions and Their Analytical Evaluations

4.1 Overload Degree and Result Quality

In this section, we explain the relationship between the Overload Degree and the result quality of the proposed approaches. We can classify the quality of results into two levels: optimal and heuristic. An optimal result means that the evacuation time is minimal. Optimal results are obtained from a huge number of combinatorial network candidates. The heavy computational load from combinatorial optimization limits the IP approach to cases of small Overload Degree, as shown in Fig. 8.

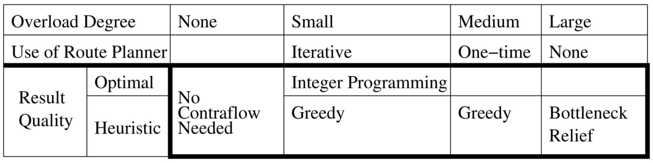
The prohibitive computational workload of achieving optimal results led us to explore effective heuristics. We designed the Greedy heuristic to meet the needs in medium-overload scenarios. As detailed in Section 5, the result quality of the Greedy heuristic compares reasonably well with the optimal result. The Bottleneck Relief heuristic is tailored to address cases of large Overload Degree.

4.2 Choice of Route Planner

In the contraflow computational framework, an evacuation route planner plays an important role in both estimating the evacuation time of a given network and providing the information of the total number of traveling units passing through each edge in the network. The estimated evacuation time is used to measure the quality of the network reconfigured by contraflow.

When we select an evacuation route planner, there is a trade-off between the result quality and runtime. An optimal evacuation route planner can generate an optimal evacuation time by performing the following three steps: creating a time-expanded network, applying a minimum-cost flow algorithm, and extracting an evacuation time. The existing minimum-cost flow solvers (for example, NETFLO [27], RELAX [6], RNET [20], and Cost Scaling (CS) [18]) are all optimal evacuation route planners. The major drawbacks of optimal evacuation route planners are their poor scalability and the requirement of prior knowledge of the upper bound of the evacuation time. These linear methods have an exponential runtime proportional to a given network size.

A heuristic evacuation route planner, by contrast, avoids these issues, often producing a close-to-optimal evacuation time with good scalability. The CCRP [30] algorithm is the only heuristic evacuation route planner available in this domain. The algorithm divides evacuees from each source into multiple groups and assigns a route and time schedule to each group based on its destination arrival time. In terms



Overload Degree		None	Small	Medium	Large
Use of Route Planner			Iterative	One−time	None
Result Quality	Optimal	No Contraflow Needed	Integer Programming		
	Heuristic		Greedy	Greedy	Bottleneck Relief

Fig. 8. Dominance zone of the proposed approaches.

of traffic assignment, CCRP is neither system optimal nor user equilibrium. However, CCRP does use a close approximation of a system-optimal approach. The heuristic gives priority to the route with the earliest destination arrival time. Even though CCRP does not produce optimal solutions in all evacuation scenarios, experiment results show that most solutions are less than 10 percent longer than the optimal evacuation time. In addition, the planner does not require the preprocessing of a given network or the upper bound of the evacuation time.

The following sections present an analytical evaluation of evacuation route planners. We use the following notations to describe the original network: p number of traveling units, n vertices, and m edges.

4.2.1 Optimal Route Planner, RELAX [6]

RELAX is a software code to solve the classic minimum cost flow problem with integer data. It is based on the relaxation method, which is designed to solve simultaneous equations by guessing a solution and reducing the errors that result by successive approximations until the errors are less than some specified amount. Bertsekas, who created the minimum flow solver RELAX, noted that there is no known polynomial complexity bound for the relaxation method [5]. Thus, we have to depend on experimental evaluation to measure the performance of the RELAX evacuation route planner in combination with the Greedy heuristic.

4.2.2 Optimal Route Planner, Cost Scaling [18]

The CS minimum cost flow algorithm combines the ideas of cost scaling, the push-relabel maximum flow method, and the relaxation method. Goldberg incorporated several heuristics (for example, price update, price refinement, arc fixing, and push look-ahead) to improve the practical performance of the CS algorithm. However, we will use the asymptotic worst-case time bound $O(n^2 m \log(nC))$ in our analysis, which is not affected by the heuristics (C: biggest cost).

As described previously, an optimal evacuation route planner runs three steps to generate an evacuation plan. First, it generates a time-expanded graph from a given network. Second, a minimum cost flow method is applied on the time-expanded graph. Third, the postprocessing of the flow history result retrieves the evacuation time. The second step is dominant in terms of runtime. G_T is the time-expanded graph built from the original network with upper bound T. The upper bound number of nodes in G_T is $N = n(T+1)$, and the upper bound number of edges is $M = (n+m)T + m - \Sigma_{(i,j)\in m}\lambda_{ij}$, where λ_{ij} denotes travel time of edge (i,j) [21]. If we assume, as is generally true, that the transportation network is sparse, with an average degree of vertices 3, we can assume that $m = 3n$. We can also assume that the maximum evacuation time T is proportional to the occupancy value p. Then, N is proportional to np, and M is also proportional to np without loss of generality. The time bound of CS is $O(N^2 M \log(NC))$ in G_T. If we combine our assumptions with the upper bound, we can acquire the following runtime: $O(N^2 M \log(NC)) = O(n^3 p^3 \log(npC))$. That is, the combination of the Greedy heuristic with CS runs superlinearly proportional to the number of nodes and traveling units.

4.2.3 Heuristic Route Planner, CCRP

The algebraic cost model of CCRP is presented in [30]. The CCRP evacuation route planner uses an iterative approach. At each iteration, the route for one group of people is chosen, and the capacities along the route are reserved. The total number of iterations is equivalent to the number of groups generated. The computation of routes for each group is performed by running the generalized Dijkstra's shortest path search. The implementation following double bucket data structures leads to an algebraic cost model of $O(p(m+2Cn))$, where C is the maximum edge weight. That is, the combination of the Greedy heuristic with CCRP runs linearly proportional to the number of nodes and evacuees.

Lemma 1. *The Greedy heuristic with a heuristic evacuation route planner is faster than the Greedy heuristic with an optimal route evacuation planner.*

Proof. In the Greedy algorithm, step 1 runs the evacuation route planner one time and is dominant in terms of runtime. Thus, a direct comparison between the optimal and heuristic evacuation route planner runtimes can prove the lemma. The optimal evacuation route planner (CS) runs in $O(n^3 p^3 \log(npC))$ and the heuristic evacuation route planner (CCRP) runs in $O(p(m+2Cn))$. By comparing the two runtimes, we can conclude that the Greedy algorithm with a heuristic evacuation route planner is faster than that with an optimal evacuation route planner when the $\frac{n^3p^2log(npC)}{m+2Cn} > 1$ relation holds, which is always true in transportation networks. □

Lemma 2. *The Bottleneck Relief heuristic is faster than the Greedy one with CCRP if* $p > \frac{9nlog^3n}{(3+2C)n}$.

Proof. The Bottleneck Relief heuristic runs in $O(m^2 log^3 n)$. The runtime of CCRP is $O(p(m+2Cn))$. By comparing the two runtimes with the assumption of a sparse transportation network $(m = 3n)$, we can conclude that the Bottleneck Relief heuristic is faster than the Greedy one with CCRP if $p > \frac{9nlog^3n}{(3+2C)n}$. We can verify the formula using the metropolitan scenario used in our experiments with a two-mile radius zone, which has 269,635 (p) occupancy, 562 (n) nodes, and 1,443 (m) edges. This data set fits the sparse network assumption $(m = 3n)$. If the parameter values are plugged in, we can observe that the formula satisfies the following: $(p = 269,635) > (\frac{9nlog^3n}{(3+2C)n} = 21,038)$. □

5 Experimental Evaluation

5.1 Experiment Setup

We implemented and evaluated the algorithms using real-world data sets. The language used was C++, and the experiments were performed on a dual-CPU Pentium III 650-MHz workstation with 2 Gbytes of memory running Linux.

5.1.1 System Design

We implemented the IP formulation on CPLEX, a mathematical programming optimizer. CPLEX is a well-known commercial optimization tool to solve IP. The Greedy heuristic uses an evacuation route planner as a plug-in external module. The two communicate via text files to exchange evacuation time and flow history information. This implementation framework gives a flexible structure to accommodate new evacuation route planners or future enhancements of existing planners. Evaluating the reconfigured contraflow network is optional and therefore not included in measuring the performance of the approaches. In the case of the Bottleneck Relief heuristic, the original network is directly used as an input because the heuristic requires only the capacity information of a given network.

5.1.2 Data Set Description

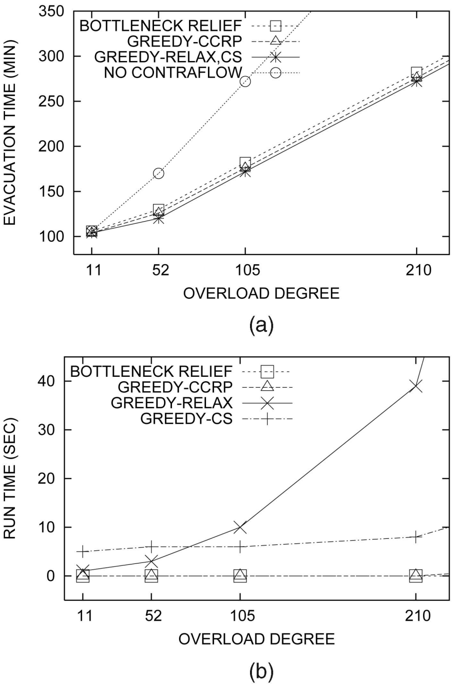
We prepared two data sets. The first is a virtual scenario of a nuclear power plant failure in Monticello, Minnesota. There are 12 cities directly affected by the failure within 10 miles of the facility and one destination shelter. This scenario is a special type of evacuation having a single destination because all evacuees should have a radioactive clearance check at the designated facility. The demographic data is based on Census 2000 population data. The total number of evacuees is about 42,000. If the given situation is represented as a graph, it contains 47 nodes with 148 edges. The graph structure is based on large edge granularity with an interstate highway (I-94) and major arterial roads. Thus, the size of the network is relatively small. The interstate highway has a larger capacity than other edges. The evacuation time with the original network configuration is 272 minutes (4 hours and 32 minutes). We created this small-size case file (in terms of number of nodes) to compare the quality of our heuristics with the optimal network configuration from IP.

The second data set was prepared with our evacuation scenario generation software. The software has the capability of specifying the size of the evacuation zone, adjusting the amount of population, changing the mode of evacuation between driving and walking, and globally adjusting the capacity of edges. With these functionalities, we were able to generate scenarios with various sizes around the Minneapolis-Saint Paul, Minnesota, metropolitan area.

The data used in the software is given as follows:

- **Map data.** This consists of TP+ (planning purpose) and Minnesota Department of Transportation basemap data (detailed geometry representation). The TP+ contains road type, road capacity, travel time, number of lanes, etc. It also contains virtual nodes as population centroids for each traffic analysis zone.
- **Population data.** This consists of Census 2000 data (nighttime estimation) and employment data (daytime estimation) but not including travelers (for example, shoppers).

We selected three different locations, as mandated by the Department of Homeland Security, with three different network sizes (that is, half-, one-, and two-mile radii). For security reasons, the specific names of the locations have been removed in this paper. The primary purpose of the second data set is to compare the results from heuristic approaches and to test the scalability due to the relatively larger network size compared to the first data set.



(a)

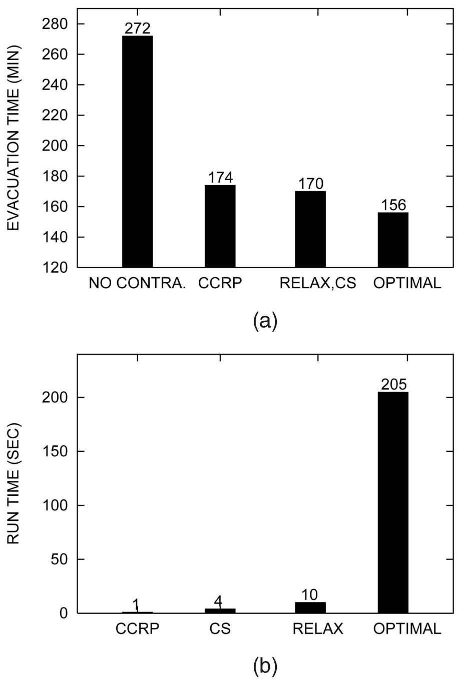
(b)

Fig. 9. Evacuation time/runtime with regard to the Overload Degree using the Monticello scenario.

5.2 Overload Degree and Result Quality

As explained previously, the Overload Degree is an important parameter in classifying the computational structure of the contraflow problem and the proposed heuristics according to the degree of computational workload. In this section, we examine the relationship between the Overload Degree and other factors such as the evacuation time and runtime of our heuristics using the Monticello data set. Fig. 9a shows the effects of the Overload Degree on evacuation time. We performed this test by changing the number of traveling units over source nodes. We can observe that the evacuation time is linearly proportional to the Overload Degree for all methods. Most heuristics showed a reduction of evacuation time by about 30 percent regardless of the Overload Degree. The Greedy heuristic with an optimal evacuation route planner (RELAX) always showed minimum evacuation time. The combination of the Greedy heuristic with a heuristic evacuation route planner (CCRP) placed in the middle. The Bottleneck Relief heuristic also showed comparable result quality with the Greedy heuristic.

Fig. 9b shows the relationship between the Overload Degree and runtime. The Greedy heuristic with RELAX (optimal evacuation route planner) has a steeply, perhaps superlinearly, increasing runtime. Greedy with CS (optimal evacuation route planner) showed a remarkably faster runtime than Greedy with RELAX as the Overload Degree increases. However, Greedy with CCRP (heuristic evacuation route planner) was the fastest (almost-zero runtime, along with the Bottleneck Relief heuristic in Fig. 9b) among the combinations of Greedy heuristics with various evacuation route planners. These results indicate that the selection of evacuation route planner is a critical design decision for scalability. The Bottleneck Relief heuristic had a constant



(a)

(b)

Fig. 10. Result quality and runtime comparison between the Greedy heuristic and IP formulation using the Monticello scenario.

runtime because it did not involve occupancy data (that is, the number of evacuees) as part of the input.

Fig. 10a shows the quality of Greedy heuristics by comparing the results from IP. First, Greedy heuristics, regardless of evacuation route planner, showed about a 40 percent reduction of evacuation time. Greedy with an optimal evacuation route planner (RELAX or CS) resulted in only a slightly better evacuation time than that with the heuristic evacuation route planner (CCRP). Second, we observe that a gap (14 minutes) exists between Greedy heuristics and optimal results. Fig. 10b shows a runtime comparison. IP resulted in a much higher runtime (205 seconds) because the IP formulation took 130,109 iterations to produce an optimal contraflow network, whereas the Greedy heuristics took only one iteration.

5.3 Choice of Route Planner and Scalability

Fig. 11 shows the convergence pattern with regard to the Degree of Contraflow using RELAX and CCRP. Although Greedy with RELAX always produced better results than that with CCRP, CCRP provided a similar result quality as RELAX, showing only a 4-minute gap in evacuation time (RELAX: 170 minutes, CCRP: 174 minutes). Both planners also showed similar convergence patterns with regard to the Degree of Contraflow. In the Monticello case, less than 10 percent of the total edges contribute to the constantly reduced evacuation time.

We also performed experiments on metropolitan scenarios to examine the convergence patterns of the Degree of Contraflow. Most evacuation times converged within 30 percent Degree of Contraflow. This means that the limited resources required to implement contraflow, such as barricade trucks and police cars, can be effectively dispatched to the appropriate locations based on the Degree of Contraflow parameter. The maximum gap in evacuation time observed between RELAX and CCRP was 32 minutes, and the minimum gap was 0 minutes. On the average, the metropolitan data sets showed a 45 percent reduction in evacuation time by contraflow from the original to the reconfigured network.

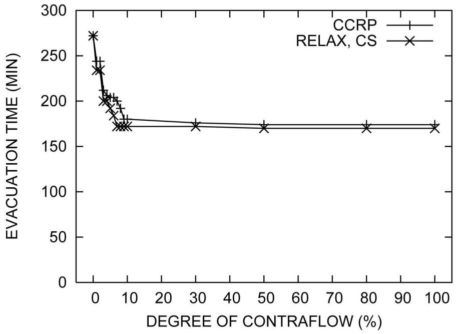


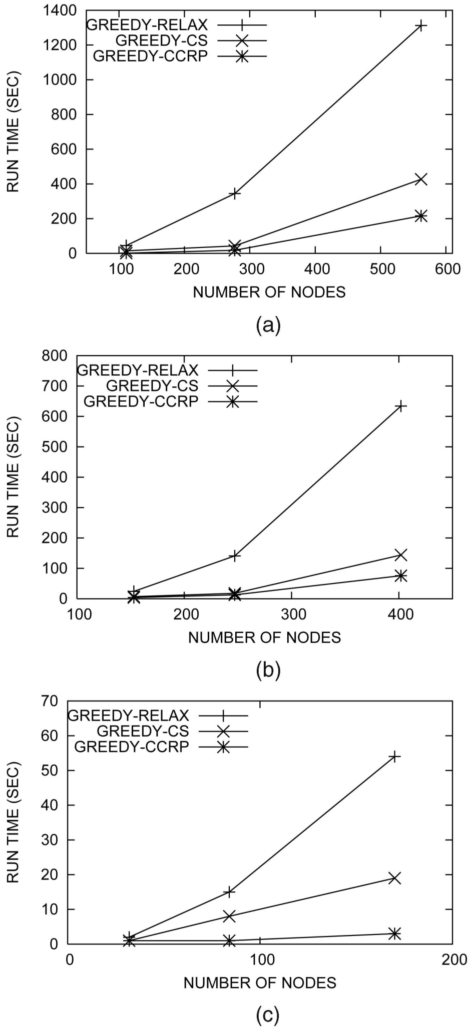
Fig. 11. Convergence pattern of the evacuation time with regard to the Degree of Contraflow using the Monticello scenario.

Fig. 12 shows the scalability of the Greedy heuristic with different evacuation route planners using metropolitan scenarios. The evacuation route planner RELAX showed a steep runtime increase. The evacuation route planner CS showed better scalability even though it produces the same result quality as RELAX. As shown in the graph, CCRP provided the best performance scalability with regard to the network size. Nowadays, evacuation at the metropolitan scale is often the issue of interest. In such cases, CCRP will play an important role in scaling our approaches to tackling huge networks.

5.4 Monticello Scenario Results and Implications for Planning

In this section, we describe two findings from the Monticello scenario that have especially important implications for evacuation route planning. The first finding is the efficiency of the computerized evacuation route planning. Fig. 13 compares a handcrafted plan with routes suggested by transportation analysts of the Department of Transportation and a plan generated by the heuristic CCRP evacuation route planner and the Greedy heuristic. The handcrafted version (Fig. 13a) results in an evacuation time without contraflow that is twice as long (554 minutes) as that generated by CCRP (276 minutes) (Fig. 13b). The main reason for the reduction in evacuation time achieved by the route planner is its ability to correctly select the direction of edges, as well as its extensive use of various routes around the destination.

A second finding is the efficiency of the computerized contraflow reconfiguration. On the network shown in Fig. 13c, we observe that 10 percent of the edges are chosen for contraflow by the Greedy heuristic. The resulting reconfigured network can further reduce the evacuation time to 180 minutes, which is 32 percent of the time required by the original handcrafted version. The 10 percent Degree of Contraflow is meaningful in that we can apply limited resources to the most congested road segment for contraflow and reserve the remaining capacity for incoming emergency traffic. In the context of transportation planning, most edges selected for contraflow in our experiments correspond to major highways with large capacity and local arterial roads around the destination. This selection scheme



(a)

(b)

(c)

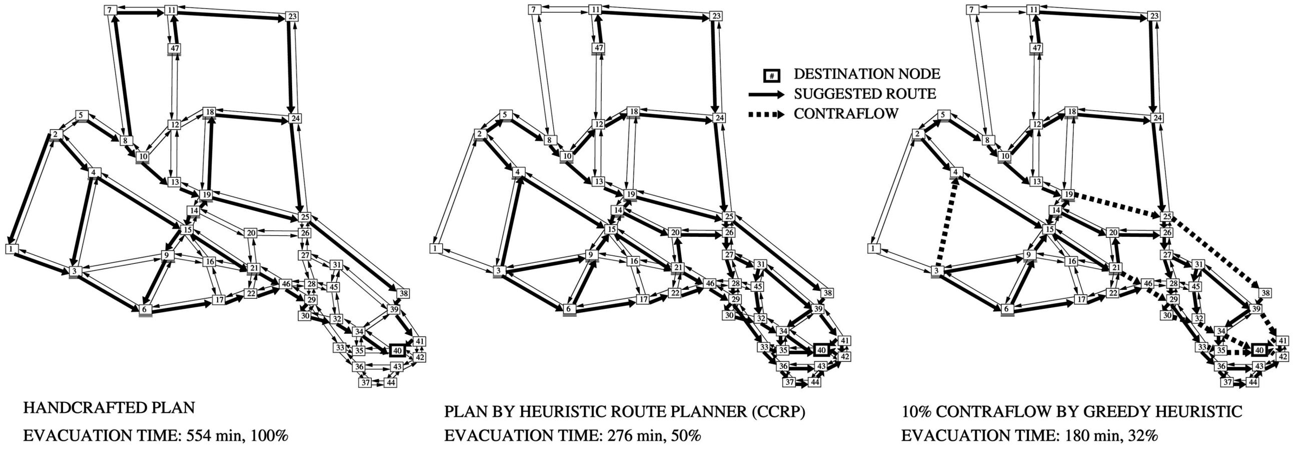
Fig. 12. Scalability with regard to the network size using metropolitan scenarios. (a) Scenario A–Runtime. (b) Scenario B–Runtime. (c) Scenario C–Runtime.

will help planners to identify and refine more efficient routes for contraflow.

6 Conclusion and Future Work

Current evacuation procedures depend heavily on the use of surface traffic through the limited capacity of road networks. From this perspective, contraflow must be seen as one of the key elements in any evacuation planned on the existing transportation infrastructure. Taking into account the nature of transportation networks, we modeled and analyzed evacuation situations using graph theory. In our model, one or more source nodes can be added, whereas existing algorithms only cover situations with a single source due to conflicts of optimal paths from different source nodes. The multiple-source and multiple-destination contraflow problem belongs to a category of NP-complete problems. Our main contribution lies in the fact that we address such a challenging contraflow problem with computational structure analysis and provide scalable heuristics with high-quality solutions. We also presented analytical and experimental evaluations. The following summarizes the two contraflow heuristics we developed:

- **Greedy heuristic.** This guarantees a promising solution quality in spite of its fast runtime. The evacuation planning software needs to be interactive due to various combinations of input parameters and evolving data sets. Thus, runtime is a critical factor when we implement a computerized contraflow planner. Our well-designed approach, based on a Greedy implementation that is tailored to contraflow problems, has some advantages over general iterative methods. With our Greedy heuristic, the number of contraflowed edges is adjustable. The scalability is superior to that of mathematical programming or simulation-based approaches.
- **Bottleneck Relief heuristic.** This is suitable for a contraflow situation with large numbers of evacuees. Although we were able to observe comparable result quality with the Greedy approach, the runtime of the Bottleneck Relief heuristic is fastest regardless of the number of traveling units.


DESTINATION NODE
SUGGESTED ROUTE
CONTRAFLOW
HANDCRAFTED PLAN
EVACUATION TIME: 554 min, 100%
PLAN BY HEURISTIC ROUTE PLANNER (CCRP)
EVACUATION TIME: 276 min, 50%
10% CONTRAFLOW BY GREEDY HEURISTIC
EVACUATION TIME: 180 min, 32%

(a) (b) (c)

Fig. 13. Handcrafted versus computerized plans. (a) Handcrafted plan evacuation time: 554 min, 100 percent. (b) Plan by heuristic route planner (CCRP) evacuation time: 276 min, 50 percent. (c) 10 percent contraflow by greedy heuristic evacuation time: 180 min, 32 percent.

Even though a contraflow operation on urban arterial roadways and long sections of interstate freeways for evacuations is accompanied by complicated issues of safety, accessibility, and cost, our proposed algorithms for simplified situations should be considerably helpful to planners designing contraflow plans because the objective of our research is to minimize the evacuation time, which is an essential part of planning.

Future work. More in-depth research is required for contraflow algorithms. Other possible methods should be examined such as the possibility of flipping a path instead of an edge. In addition, we need to explore the application of queuing theory [1], [4], [9] and search techniques in the artificial intelligence field [19], [33], [35] to the contraflow problem. We will use an evacuation route planner based on microscopic simulation to see how detailed congestion phenomena affect the choice of edges to be reversed. Inbound traffic demand should be considered. Network capacity should be preempted for emergency vehicles, traffic officers, or firefighters. Partial lane reversal and time-dependent capacity-varying edges need to be incorporated in the modeling. The quantitative values of the Overload Degree—for example, the practical range of medium Overload Degree—need to be established and refined. We will develop a more realistic congestion index formula using the fundamental diagram between flow, density, and speed frequently used in the traffic operations area. Finally, the time-dependent nature of traffic flow can be addressed in the evacuation route planner.

ACKNOWLEDGMENTS

The authors are particularly grateful to the members of the Spatial Database Research Group at the University of Minnesota for their helpful comments and valuable discussions. The authors' evacuation research members, including Qingsong Lu and Betsy George, gave them especially valuable comments. The ArcGIS Network Analyst and geodatabase teams at ESRI have given the authors a chance to implement the ideas in this paper on their GIS development framework. Kim Koffolt and ESRI helped the authors improve the readability of the paper. They are thankful for the use of the program RelaxIV, developed by the CRIFOR research group, and CS, developed by A. Goldberg. This work was supported by a grant (Contract 81655) from the Minnesota Department of Transportation.

REFERENCES

[1] I. Adan, *Queueing Theory.* Eindhoven Univ. of Technology, 2001.

[2] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin, *Network Flows: Theory, Algorithms, and Applications.* Prentice Hall, 1993.

[3] M. Ben-Akiva, "Development of a Deployable Real-Time Dynamic Traffic Assignment System: DynaMIT and DynaMIT-P User's Guide," technical report, Massachusetts Inst. of Technology, 2002.

[4] A.W. Berger, L.M. Bregman, and Y. Kogan, "Bottleneck Analysis in Multiclass Closed Queueing Networks and Its Application," *Queueing Systems,* vol. 31, no. 3-4, pp. 217-237, 1999.

[5] D. Bertsekas, personal communication by E-mail, available upon request, 2006.

[6] D. Bertsekas and P. Tseng, "Relax-IV: A Faster Version of the Relax Code for Solving Minimum Cost Flow Problems," Technical Report P-2276, Laboratory for Information and Decision Systems, Massachusetts Inst. of Technology, Cambridge, Mass., 1994.

[7] R.J. Caudill and N.M. Kuo, "Development of an Interactive Planning Model for Contraflow Lane Evaluation," *Transportation Research Record, Urban Traffic Systems,* vol. 906, no. 7, pp. 47-54, 1983.

[8] X. Chen, J. Meaker, and F. Zhan, "Agent-Based Modeling and Analysis of Hurricane Evacuation Procedures for the Florida Keys," *Natural Hazards,* vol. 38, pp. 321-338, 2006.

[9] W.C. Cheng and R.R. Muntz, "Optimal Routing for Closed Queueing Networks," *Proc. 14th IFIP WG 7.3 Int'l Symp. Computer Performance Modelling, Measurement and Evaluation (Performance '90),* pp. 3-17, 1990.

[10] T.J. Cova and J.P. Johnson, "A Network Flow Model for Lane-Based Evacuation Routing," *Transportation Research Part A: Policy and Practice,* vol. 37, pp. 579-604, 2003.

[11] J.G. Doheny and J.L. Fraser, "Mobedic—A Decision Modeling Tool for Emergency Situations," *Expert Systems with Applications,* vol. 10, pp. 17-27, 1996.

[12] M. Ebihara, A. Ohtsuki, and H. Iwaki, "A Model for Simulating Human Behavior during Emergency Evacuation Based on Classificatory Reasoning and Certainty Value Handling," *Microcomputers in Civil Eng.,* vol. 7, pp. 63-71, 1992.

[13] Federal Highway Administration, *Catastrophic Hurricane Evacuation Plan Evaluation,* http://www.onewayflorida.org/, 2007.

[14] Florida Department Transportation, *State of Florida Contraflow Plan,* http://www.onewayflorida.org/, 2007.

[15] G. Ford, R. Henk, and P. Barricklow, "Interstate Highway 37 Reverse Flow Analysis—Technical Memorandum," technical report, Texas Transportation Inst., 2000.

[16] L.R. Ford and D.R. Fulkerson, *Flows in Networks.* Princeton Univ. Press, 1962.

[17] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP Completeness.* W.H. Freeman, 1979.

[18] A.V. Goldberg, "An Efficient Implementation of a Scaling Minimum-Cost Flow Algorithm," *J. Algorithms,* vol. 22, pp. 1-29, 1997.

[19] R. Greiner, "Probabilistic Hill-Climbing: Theory and Applications," *Proc. Ninth Canadian Conf. Artificial Intelligence,* pp. 60-67, 1992.

[20] M.D. Grigoriadis, "An Efficient Implementation of the Network Simplex Method," *Math. Programming Study,* vol. 26, pp. 83-111, 1986.

[21] H.W. Hamacher and S.A. Tjandra, "Mathematical Modeling of Evacuation Problems: State of the Art," *Pedestrian and Evacuation Dynamics,* pp. 227-266, 2001.

[22] G. Hamza-Lup, K.A. Hua, M. Le, and R. Peng, "Enhancing Intelligent Transportation Systems to Improve and Support Homeland Security," *Proc. Seventh IEEE Int'l Conf. Intelligent Transportation Systems (ITSC),* 2004.

[23] E. Hoel, W. Heng, and D. Honeycutt, "High Performance Multimodal Networks," *Proc. Ninth Int'l Symp. Advances in Spatial and Temporal Databases (SSTD '05),* pp. 308-327, 2005.

[24] S. Hoogendoorn and P. Bovy, "State-of-the-Art of Vehicular Traffic Flow Modelling," *Proc. Institution of Mechanical Engineers,* vol. 215, no. 1, pp. 283-303, 2001.

[25] M. Jha, K. Moore, and B. Pashaie, "Emergency Evacuation Planning with Microscopic Traffic Simulation," *Transportation Research Record: J. Transportation Research Board,* vol. 1886, pp. 40-48, 2004.

[26] D. Karger and C. Stein, "An $O(n^2)$ Algorithm for Minimum Cuts," *Proc. 25th ACM Symp. Theory of Computing (STOC '93),* pp. 757-765, 1993.

[27] J.L. Kennington and R.V. Helgason, *Algorithm for Network Programming.* John Wiley & Sons, 1980.

[28] S. Kim and S. Shekhar, "Contraflow Network Reconfiguration for Evacuation Planning: A Summary of Results," *Proc. 13th ACM Symp. Advances in Geographic Information Systems (GIS '05),* pp. 250-259, 2005.

[29] T. Litman, "Lessons from Katrina and Rita: What Major Disasters Can Teach Transportation Planners," *J. Transportation Eng.,* vol. 132, no. 1, pp. 11-18, 2006.

[30] Q. Lu, B. George, and S. Shekhar, "Capacity Constrained Routing Algorithms for Evacuation Planning: A Summary of Results," *Proc. Ninth Int'l Symp. Spatial and Temporal Databases (SSTD '05),* pp. 291-307, 2005.

[31] H.S. Mahmassani, H. Sbayti, and X. Zhou, "DYNASMART-P Version 1.0 User's Guide," technical report, Maryland Transportation Initiative, Univ. of Maryland, 2004.

[32] "1998 Performance of Regional High-Occupancy Vehicle Facilities on Freeways in the Washington Region," *Analysis of Person and Vehicle Volumes,* Metropolitan Washington Council of Gov't, 1999.

[33] S. Nahar, S. Sahni, and E. Shragowitz, "Simulated Annealing and Combinatorial Optimization," *Proc. 23rd ACM/IEEE Design Automation Conf. (DAC '86),* pp. 293-299, 1986.

[34] M. Pidd, F. Silva, and R. Eglese, "Simulation Model for Emergency Evacuation," *European J. Operational Research,* vol. 90, no. 3, pp. 413-419, 1996.

[35] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach,* second ed. Prentice Hall, 2003.

[36] *Federal Response to Hurricane Katrina: Lessons Learned,* http://www.whitehouse.gov/reports/katrina-lessons-learned/, The White House, 2006.

[37] G. Theodoulou, "Contraflow Evacuation on the Westbound I-10 out of the City of New Orleans," master's thesis, Louisiana State Univ., 2003.

[38] G. Theodoulou and B. Wolshon, "Alternative Methods to Increase the Effectiveness of Freeway Contraflow Evacuation," *Transportation Research Record: J. Transportation Research Board,* vol. 1865, pp. 48-56, 2004.

[39] H. Tuydes and A. Ziliaskopoulos, "Network Re-Design to Optimize Evacuation Contraflow," Technical Report 04-4715, Proc. 83rd Ann. Meeting of the Transportation Research Board, 2004.

[40] H. Tuydes and A. Ziliaskopoulos, "Tabu-Based Heuristic for Optimization of Network Evacuation Contraflow," Technical Report 06-2022, Proc. 85th Ann. Meeting of the Transportation Research Board, 2006.

[41] B. Wolshon, "One-Way-Out: Contraflow Freeway Operation for Hurricane Evacuation," *Natural Hazards Rev.,* vol. 2, no. 3, pp. 105-112, 2001.

[42] B. Wolshon, E. Urbina, and M. Levitan, "National Review of Hurricane Evacuation Plans and Policies," technical report, Hurricane Center, Louisiana State Univ., Baton Rouge, 2002.

Sangho Kim received the master's degree and PhD degree (in 2007) in computer science from the University of Minnesota. He is currently a software developer with the Geodatabase team at ESRI. His research interests include issues in geodatabases, spatial network problems, evacuation routing problems, optimizations, and geographic information systems.

Shashi Shekhar is a McKnight distinguished university professor at the University of Minnesota. He is the coauthor of a popular textbook on spatial databases and is serving on the NAS/NRC Mapping Science Committee and as a coeditor in chief of *GeoInformatica*. He was elected an IEEE fellow for contributions to spatial database storage methods, data mining, and geographic information systems.

Manki Min received the PhD degree in computer and information sciences from the University of Minnesota in 2004. He is currently an assistant professor in the Electrical Engineering and Computer Science Department, South Dakota State University (SDSU). Before joining SDSU, he was in a postdoctoral research associate at the University of Florida, working with Distinguished Professor Dr. Panos M. Pardalos. His current research interests include the design and analysis of optimization techniques and heuristics with main applications to various network topologies. He is a member of the IEEE.