

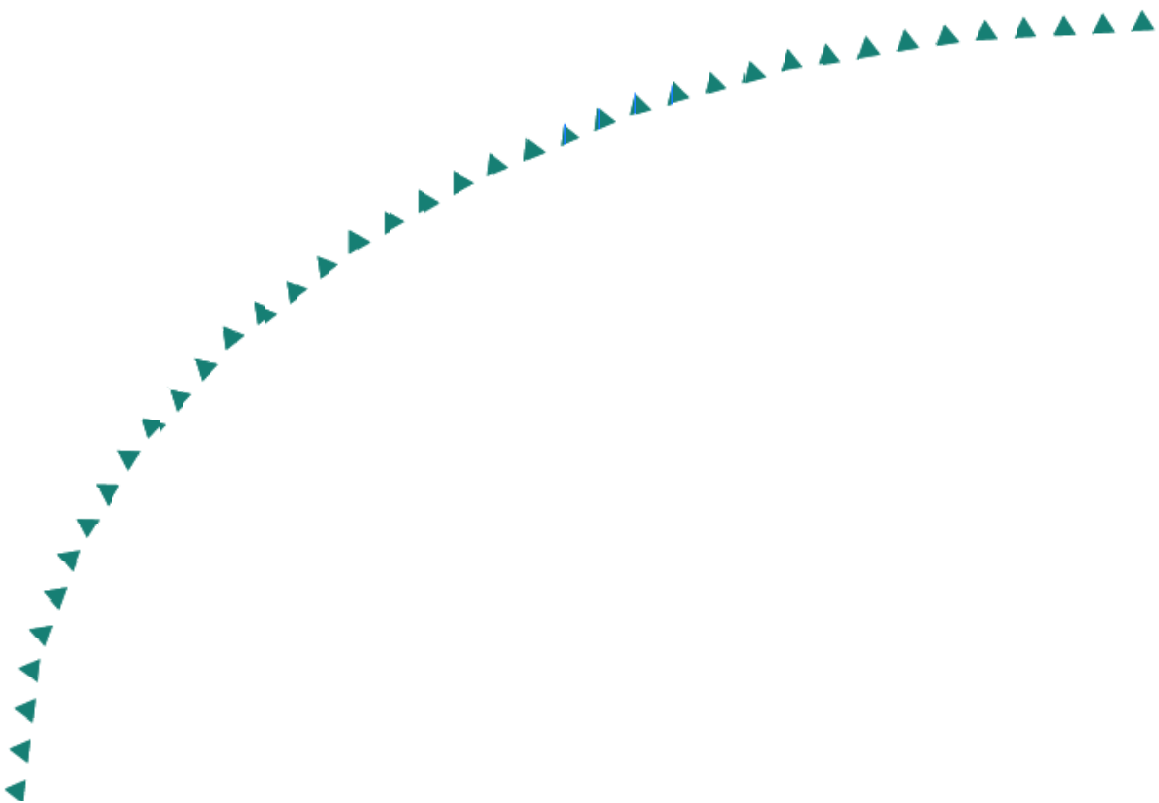
2006-21

Final Report

Contraflow Transportation Network
Reconfiguration for Evacuation
Route Planning



Research



Technical Report Documentation Page

1. Report No. MN/RC-2006-21	2.	3. Recipients Accession No.	
4. Title and Subtitle Contraflow Transportation Network Reconfiguration for Evacuation Route Planning		5. Report Date June 2006	
		6.	
7. Author(s) Shashi Shekhar, Sangho Kim		8. Performing Organization Report No.	
9. Performing Organization Name and Address Department of Computer Science and Engineering University of Minnesota 4-192 EE-CSci Building 200 Union Street SE Minneapolis, MN 55455		10. Project/Task/Work Unit No.	
		11. Contract (C) or Grant (G) No. (C) 81655 (wo) 97	
12. Sponsoring Organization Name and Address Minnesota Department of Transportation Research Services Section 395 John Ireland Boulevard Mail Stop 330 St. Paul, Minnesota 55155		13. Type of Report and Period Covered Final Report June 2004 – June 2006	
		14. Sponsoring Agency Code	
15. Supplementary Notes http://www.lrrb.org/PDF/200621.pdf			
16. Abstract (Limit: 200 words) <p>Given a transportation network having source nodes with evacuees and destination nodes, we want to find a contraflow network configuration, i.e., ideal direction for each edge, to minimize evacuation time. Contraflow is considered a potential remedy to reduce congestion during evacuations in the context of homeland security and natural disasters (e.g., hurricanes). This problem is computationally challenging because of the very large search space and the expensive calculation of evacuation time on a given network. To our knowledge, this paper presents the first macroscopic approaches for the solution of contraflow network reconfiguration incorporating road capacity constraints, multiple sources, congestion factor, 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 datasets. Experimental results show that our contraflow approaches can reduce evacuation time by 40% or more.</p>			
17. Document Analysis/Descriptors Contraflow, Transportation, Optimization, Evacuation Route Planning		18. Availability Statement No restrictions. Document available from: National Technical Information Services, Springfield, Virginia 22161	
19. Security Class (this report) Unclassified	20. Security Class (this page) Unclassified	21. No. of Pages 48	22. Price

Contraflow Transportation Network Reconfiguration for Evacuation Route Planning

Final Report

Prepared by:

Shashi Shekhar

Sangho Kim

Department of Computer Science and Engineering
University of Minnesota

June 2006

Published by:

Minnesota Department of Transportation
Research Services Section
395 John Ireland Boulevard, MS 330
St. Paul, Minnesota 55155-1899

This report represents the results of research conducted by the authors and does not necessarily represent the views or policies of the Minnesota Department of Transportation and/or the Center for Transportation Studies. This report does not contain a standard or specified technique. The authors and the Minnesota Department of Transportation and/or Center for Transportation Studies do not endorse products or manufacturers. Trade or manufacturers' names appear herein solely because they are considered essential to this report

Acknowledgements

We are grateful to the Minnesota Department of Transportation for supporting this project through Contract No. 81655. We are particularly grateful to members of the Spatial Database Research Group at the University of Minnesota for their helpful comments and valuable discussions. Our evacuation research members including Qingsong Lu and Betsy George gave us especially valuable comments. Kim Koffolt helped us improve the readability of our paper. We are thankful for the use of quotations on hurricane experience and disaster classification table by T. Litman, the program RelaxIV developed by the CRIFOR research group and CS by A. Goldberg.

Table of Contents

Chapter 1. Introduction	1
1.1 Motivation.....	2
1.2 Problem Formulation.....	4
1.3 Problem Hardness	5
1.4 Related Work and Our Contribution	6
1.5 Scope and Outline of the Paper.....	8
Chapter 2. Computational Framework and Proposed Approaches	9
2.1 Computational Framework	9
2.2 Greedy Heuristic	10
2.3 Bottleneck Relief Heuristic for Large Overload Degree	13
Chapter 3. Design Decisions and Their Analytical Evaluations	15
3.1 Overload Degree and Result Quality.....	15
3.2 Choice of Route Planner	15
3.2.1 Optimal Route Planner: RELAX.....	16
3.2.2 Optimal Route Planner: Cost Scaling (CS).....	16
3.2.3 Heuristic Route Planner: CCRP	17
Chapter 4. Experimental Evaluation	18
4.1 Experiment Setup.....	18
4.1.1 System Design	18
4.1.2 Dataset Description.....	18
4.2 Overload Degree and Result Quality.....	20
4.3 Choice of Route Planner and Scalability	21
4.4 Monticello Scenario Results and Implications for Planning	23

Chapter 5. Conclusion and Future Work 24

References 25

Appendix A. Optimal Contraflow Solution using Integer Programming

Appendix B. How Hard Is Contraflow Problem?

Appendix C. Evacuation Scenario Generation Software

Appendix D. Summary of Metro Evacuation Dataset

Appendix E. How to Measure Evacuation Time?

List of Tables

Table 1.1	Different Types of Disasters Present Different Types of Evacuation Properties.....	3
Table 2.1	Algorithm Greedy.....	11
Table 2.2	Sorted Congestion Index from Step 3 in Figure 2.3.....	12
Table 2.3	Algorithm Bottleneck Relief.....	13

List of Figures

Figure 1.1	Hurricane Rita evacuation required contraflow on Interstate 45. Notice that traffic on both sides of I-45 is going north (source: dallasnews.com).....	3
Figure 1.2	Center Two Reversible Lanes for Peak Period HOV-3 Vehicles of I-95 around Washington D.C. Area (roadtothefuture.com).....	4
Figure 1.3	Graph Representation of a Simple Evacuation Situation and Two Following Contraflow Configuration Candidates.....	5
Figure 1.4	Simple Evacuation Network with Bottleneck Capacity 3.....	6
Figure 2.1	Overload Degree and Proposed Approaches.....	9
Figure 2.2	Use of Route Planner in the Framework.....	10
Figure 2.3	Example of Greedy Algorithm.....	12
Figure 2.4	Example of Bottleneck Relief Heuristic.....	14
Figure 3.1	Dominance Zone of Proposed Approaches.....	15
Figure 4.1	Monticello Nuclear Power Plant Case Map.....	19
Figure 4.2	Evacuation time / Runtime with regard to Overload Degree using Monticello Scenario.....	20
Figure 4.3	Result Quality and Runtime Comparison Between Greedy heuristic and IP Formulation using Monticello Scenario.....	21
Figure 4.4	Convergence Pattern of Evacuation Time with regard to Degree of Contraflow using Monticello Scenario.....	21
Figure 4.5	Convergence Pattern of Evacuation Time with regard to Degree of Contraflow using Metropolitan Scenario.....	22
Figure 4.6	Scalability with regard to Network Size using Metropolitan Scenario.....	22
Figure 4.7	Handcrafted vs. Computerized Plans.....	23

Executive Summary

Contraflow, or lane reversals, has been discussed as a potential remedy to solve tremendous congestion during evacuation process by increasing outbound evacuation route capacity. We can more formally define contraflow network reconfiguration problem as follows. Given a transportation network having source nodes with evacuees and destination nodes, we want to find a contraflow network configuration, i.e., ideal direction for each edge, to minimize evacuation time. Finding the optimal contraflow network configuration is considerably challenging because we may have to enumerate combinations of edge (i.e., road segment) directions and compare those by calculating evacuation time. The difficulty originates from the combinatorial nature of the problem.

In the previous work, evacuation situation was modeled as a single source problem, which reduced the solution to overlaying multiple shortest paths. In other literature, contraflow problem was approached by mathematical optimization, which showed scalability limitation. To our knowledge, this paper presents the first macroscopic approaches for the solution of contraflow network reconfiguration incorporating road capacity constraints, multiple sources, congestion factor, and scalability.

As part of our research to address the challenges of evacuation route planning, we introduce the notion of Overload Degree to classify the computational structure of the contraflow reconfiguration problem by the ratio of the number of evacuees to bottleneck capacity of the transportation network. We propose heuristics for determining ideal direction of edges in transportation network for evacuation. The Greedy heuristic runs an evacuation route planner to see 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 experimental evaluation, we prepared real world datasets to test the performance and scalability of the approaches. Experimental results and case studies show that the proposed approaches can reduce 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. Our contraflow heuristics tackling the congestion of bottlenecks on evacuation network are scalable to handle an order of larger magnitude scenarios than those used in previous works.

Chapter 1

Introduction

Efficient evacuation route planning is currently an issue of major importance due to the increasing risks both from terrorist attacks and natural disasters. For transportation system planners, the main issue has been colossal 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 more accurate estimate of evacuation time [18].

Contraflow, or lane reversals, has been discussed as a potential remedy to solve such tremendous congestion by increasing outbound evacuation route capacity[25, 26]. Today, 11 of the 18 coastal states in the U.S. frequently threatened by hurricanes consider the use of contraflow as part of their evacuation strategy [25]. Although contraflow is primarily important for evacuations, its applications are not limited to emergencies. The two center lanes of the highway system in Washington, D.C. are used in reverse laning fashion to efficiently control capacity for morning and evening commuter peak time [3,20]. The utilization of contraflow after football games is another typical example of a single source, multiple destinations situation.

The contraflow problem for evacuation can be defined as follows. A transportation network is given with multiple sources and destinations. Each source has initial occupancy. Each directed road segment connecting two nodes has capacity and travel time. We are interested in finding a reconfigured network identifying the ideal direction for each edge to minimize evacuation time by reallocating the available capacity using lane reversals. Finding the optimal contraflow network configuration is considerably challenging because we may have to enumerate combinations of edge (i.e., road segment) directions and compare those by calculating evacuation time. The difficulty originates from the combinatorial nature of the problem.

Hamza-Lup and Hua proposed algorithms to tackle the contraflow problem [12]. In their evacuation modeling, an evacuation zone consists of a single source and multiple destinations. Their solution is reduced to finding the optimal paths from the single source to the destinations and overlaying them. This planning approach does not take capacity of road network into account. Tuydes and Ziliaskopoulos designed a mesoscopic contraflow network model based on a dynamic traffic assignment method [23]. In their modeling, a discrete network was generated using time interval and cell connectors. Their solution is subject to the problem of mathematical optimization, however, and thus they have not shown scalable experiments (i.e., network with less than 50 nodes). In addition, their Tabu-based heuristic approach was search-based iterative optimization technique, limiting their experiments of urban evacuation to road networks with high granularity of road segments [24].

As part of our research to address the challenges of evacuation route planning, we introduce the notion of Overload Degree to classify the computational structure of the contraflow reconfiguration problem by the ratio of the number of evacuees to bottleneck capacity of the transportation network. We propose heuristics for determining ideal direction of edges in transportation network for evacuation. The Greedy heuristic runs an evacuation route planner to

see 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 experimental evaluation, we prepared real world datasets to test the performance and scalability of the approaches. Experimental results and case studies show that the proposed approaches can reduce evacuation time by 40% or more. In addition, we present findings with important implications for planners and first responders as they prepare contraflow evacuation schemes. Unlike previous work, our evacuation modeling can handle a network with edge capacity constraints and multiple sources [12]. Our contraflow heuristics tackling the congestion of bottlenecks on evacuation network are scalable to handle an order of larger magnitude scenarios than those used in previous works [23,24].

1.1 Motivation

Evacuation route planning has become a topic of critical importance due to the recent September 11 terrorist attacks and catastrophic hurricanes that required large scale evacuations across the U.S. In 2005, two major Hurricanes, Katrina and Rita, hit the southeastern part of the United States and caused severe damage across several coastal states [13]. Evacuations for the two hurricanes played out in two very different ways. In the case of Katrina, many evacuees in the affected area did not have reliable personal transportation and were not willing to leave their homes. However, the opposite occurred in the case of Rita. A greater than expected number of evacuees followed the evacuation order with their personal vehicles. The following are quoted observations of the traffic problems that occurred during the Rita evacuation [18].

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 4 and 5 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 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 I-45 (Figure 1.1) 290 and I-10. But by mid-afternoon, with traffic immobile on 290, the plan was dropped, stranding many and prompting other 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 were able to observe inefficient use of road capacity and the effects from the ill-planned contraflow, resulting in disorganized movement of people [18]. They listed failure to use contraflow lanes and road shoulders for evacuation traffic as one of the planning problem lessons learned from Katrina and Rita.

Table 1.1 presents various types of disasters and their properties. According to Litman, evacuation route plans should take into account the geographic scale and length of warning [18].

Contraflow preparedness is most appropriate for disasters with large geographic scale and long warning time, which gives responders time to dispatch resources and establish reversed lanes. Thus, hurricane and flooding are the most appropriate candidates to apply contraflow plans before disaster. Other types of disasters with relatively short warning time should consider contraflow after disaster.

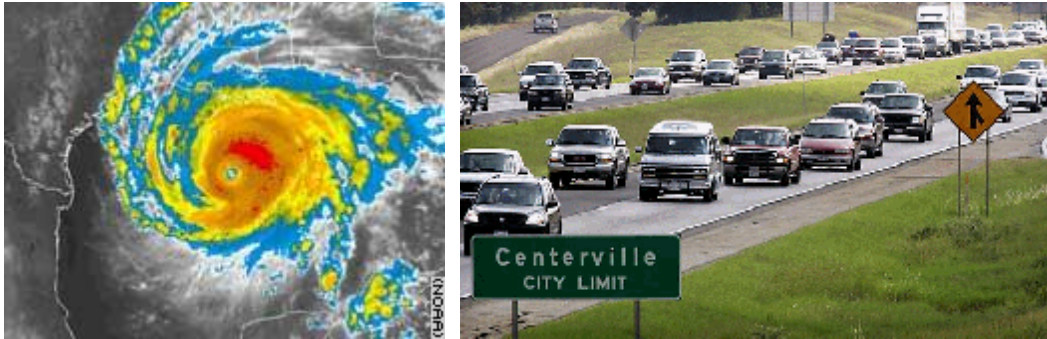


Figure 1.1. Hurricane Rita evacuation required contraflow on Interstate 45. Notice that traffic on both sides of I-45 is going north (source: dallasnews.com)

Table 1.1. Different Types of Disasters Present
Different Types of Evacuation Properties (source: [18])

Type of Disaster	Geographic Scale	Warning	Contraflow Before	Contraflow After
Hurricane	Very large	Days	√	√
Flooding	Large	Days	√	√
Earthquake	Large	None		√
Tsunami	Very large	Short		√
Radiation/toxic Release	Small to large	Sometimes		√

Although it is a subject of recent dramatic interest, contraflow has other more minor 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 2-lane roadways for peak-period HOV-3 vehicles (Figure 1.2) [3, 20]. The reversible lane system has been reported to provide significant savings in travel time. The Washington system still has room for improvement by identifying critical road segments affecting the entire system performance and efficient operating time zone to avoid waste of resources. A second application of contraflow is common during special events when all lanes are reversed to accommodate outbound traffic at the end of a sports game or concert. This is a special case of contraflow having a single source with multiple destinations.



Figure 1.2. Center Two Reversible Lanes for Peak Period HOV-3 Vehicles of I-95 around Washington D.C. Area (roadtothefuture.com)

1.2. Problem Formulation

We understand the evacuation route planning as a process to remove residents in a dangerous area to safe places as quickly as possible. It is necessary to represent the situation with a mathematical graph structure. Let $G(N, E)$ be a directed network with N , the set of nodes, and E , the set of edges. Each node has an initial occupancy value, that is, the number of residents to evacuate, and a node capacity. Each edge also has an edge capacity, constant travel time, and an initial direction. The evacuation situation can have multiple source nodes and destination nodes. Evacuation time is defined as a period from the moment when a first evacuee leaves a source node to the moment when a last evacuee arrives at a destination node. We want to find a reconfigured network by contraflow with the objective of minimizing evacuation time. 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 (i.e., desired direction for each edge)

Objective: Minimize evacuation time.

Constraint:

1. Travel time and capacity are constant.
2. Direction of each edge can be flipped to allow contraflow.
3. Edge is the smallest unit of contraflow.

Figure 1.3 illustrates a simple evacuation situation of a transportation network. Suppose that each node represents a city with initial occupancy and its capacity, as shown in Figure 1.3(a). City A has 40 people and also capacity 40. Nodes A and C are modeled as source nodes, while node E is modeled as a destination node (e.g., shelter). Each edge represents a road between two cities with travel time and its capacity. For example, a highway segment between cities A and B has travel time 1 and capacity 3. If we assume that a time unit is 5 minutes, it takes 5 minutes for evacuees to travel from A to B and a maximum of 3 evacuees can simultaneously travel through the edge. Nodes B and D have no initial occupancy and only serve as transshipment nodes. The evacuation

time of the original network in Figure 1.3(a) is 22. Details of how to measure evacuation time are presented in Appendix E.

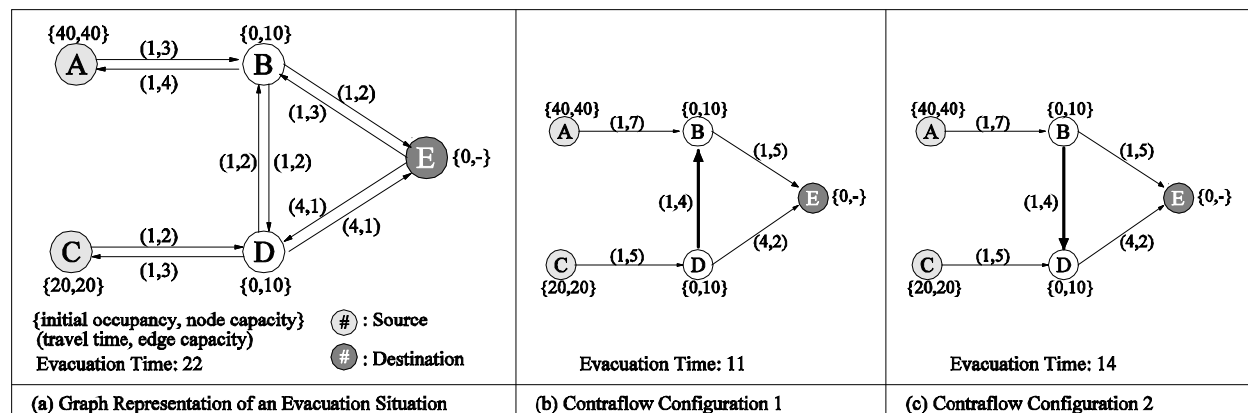


Figure 1.3. Graph Representation of a Simple Evacuation Situation and Two Following Contraflow Configuration Candidates

Figure 1.3(b) and (c) illustrates two possible contraflow configurations based on the original graph. All the two-way edges used in the original configuration are merged by capacity and directed in favor of increasing outbound evacuation capacity. There are two candidate configurations that differ in the direction of edges between nodes B and D. If we measure the evacuation times of both configurations, the configuration in Figure 1.3(b) has evacuation time 11, while the configuration in Figure 1.3(c) has evacuation time 14. We can observe that both configurations not only reduce, but also differ in, evacuation time. Even though the time difference is just 3 in this example, the difference may be significantly different in the case of a complicated real network. This example illustrates the importance of choice among possible network configurations. Moreover, we have to know that there are critical edges affecting the evacuation time, such as edge (B, D) in Figure 1.3.

We made two assumptions in our evacuation modeling. First, we assume that edge travel time and capacity are constant. In reality, travel time of an edge is not fixed as constant, but may be density dependent. Second, we only consider reversing all lanes for simplicity. In an actual implementation of contraflow, it is possible to reverse some portion of lanes.

1.3 Problem Hardness

Due to the combinatorial nature of the contraflow problem, acquiring the optimal solution becomes considerably challenging as the size of the network increases. Theoretical analysis regarding the NP-hardness of the contraflow problem is presented in Appendix B. To address such a challenging problem with regard to its size, we need to define a parameter to classify the computational structure of the problem. The number of evacuees and capacity of a given network are two critical factors affecting the computational structure. In general, the evacuation time and computational workload increase as the number of evacuees increases. The capacity is inversely proportional to the evacuation time as well as computational workload. With these notions, we introduce the term "Overload Degree", which we define as follows.

Definition 1 (Overload Degree):

Overload Degree = Number of Evacuees / Bottleneck Capacity Without Contraflow

'Bottleneck Capacity Without Contraflow' refers to a minimum cut value (or, maximum flow value) of a given network configuration without contraflow. The hardness of the contraflow problem is a function of Overload Degree. For a small Overload Degree, where the value is around a single digit, we can consider mathematical programming, search-based approaches, or microscopic simulation to produce optimal results. For a medium Overload Degree, we definitely need heuristic approaches to balance the result quality and reasonable amount of computational workload. For a large Overload Degree, we can apply simple heuristics by ignoring the transitional part of the evacuation process.

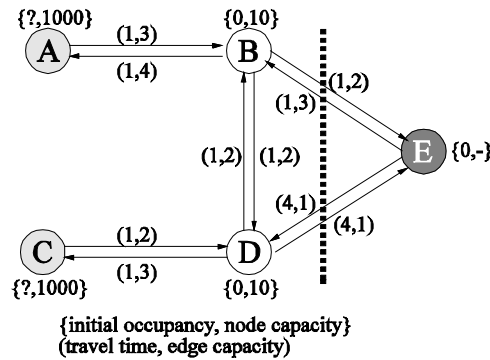


Figure 1.4. Simple Evacuation Network with Bottleneck Capacity 3

Figure 1.4 is the same example of evacuation network in Figure 1.3, but the node capacities of A and C are increased up to 1000. The dotted line is a bottleneck of this network, separating the source nodes and the destination nodes. The value of the bottleneck capacity is 3 by adding the capacity of edge (B,E) (i.e., from B to E) and (D,E). Suppose node A has occupancy 2 and C has 1. We do not need contraflow because current bottleneck capacity is enough to handle the small number of evacuees. In this case, the Overload Degree is 1. As the number of evacuees increases (e.g., > 3), current bottleneck capacity becomes insufficient. We start thinking of contraflow to reduce evacuation time. The computational workload accordingly becomes heavy to calculate the scheduling of the large number of evacuees. Suppose the nodes A and C have 1000 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 to get the notion of the problem size of a given network to apply contraflow.

1.4 Related Work and Our Contribution

The material and literature on evacuations in general and the contraflow problem in particular have been published from various domains including social and behavioral sciences, transportation, and mathematics [4, 5]. A survey of evacuation issues and contraflow revealed that planners have no recognized standards or guidelines for the design, operation, and location of contraflow segments [26]. Many states threatened by hurricanes and considering contraflow plans were dependent on past evacuation experiences. Litman identified planning problems of Hurricanes Katrina and Rita and specifically criticized unplanned contraflow orders and failure

to use contraflow lanes and road shoulders where they were available during the evacuation procedures of Hurricane Rita [18].

Past papers and Department of Transportation reports have mainly tackled the managerial and operational aspects of contraflow such as signal control, merging and cost [6, 21, 25, 26]. When planners design network configuration for evacuation scenarios, they mainly depend on empirical guesses. Such handcrafted contraflow plans have revealed that they are neither efficient to find critical road segments of contraflow nor flexible to accommodate various variables.

Hamza-Lup et al. introduced two different contraflow algorithms from a computer science perspective, one is based on a multicast routing problem and the other based on breadth-first graph traversal [12]. These algorithms can handle only a single-coordinated incident due to 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 evacuees is finite, or road capacities are constrained, or specific destination nodes are prescribed, or evacuees are spread over many locations.

Tuydes and Ziliaskopoulos proposed a mesoscopic contraflow network model based on a dynamic traffic assignment method [23]. 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 to address capacity reversibility optimization [24]. Their solutions required a considerable number of iterations, thus limiting their input to small networks in our proposed computational framework, (networks categorized by small Overload Degree).

Theodoulou and Wolshon used CORSIM microscopic traffic simulation to model the freeway contraflow evacuation around New Orleans [22]. With the help of a micro scale traffic simulator, they were able to suggest alternative contraflow configurations at the level of entry and termination points. However, their microscopic simulation model requires a labor intensive network coding using aerial photographs and significant running time for each scenario, making it difficult to take advantage of spatial databases or easily compare alternative configurations. Evacuation route plannings with other microscopic traffic simulators (e.g., MITSIMLab [14]) have shown similar limitations. In our computational framework, microscopic simulations are only applicable to transportation networks with a small Overload Degree.

Our Contributions: Previously, Kim and Shekhar proposed two heuristics for contraflow planning [17]. One heuristic named Flip High Flow Edge (FHFE) is based on a greedy algorithm with a flow history of edges. The FHFE generates a sub-optimal contraflow plan without iteration. The other heuristic is based on a simulated annealing optimization technique. Due to the searching property (i.e., 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 with 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, i.e., ratio of number of evacuees to the bottleneck capacity of a transportation network. For a small Overload Degree, various iterative methods are available such as mathematical programming, search-based optimization techniques or microscopic simulation. Such techniques, however, have suffered scalability problems due to their expensive computation. Therefore, we present a Greedy heuristic designed to handle scenarios with a significantly large population and network size. Asymptotically fast runtime of the Greedy heuristic expands its applicability to metropolitan evacuation scenarios. It also has a flexible algorithm structure by using an evacuation route planner as a plug-in module, thus leaving room for improvements as faster evacuation route planners evolve in the domain of evacuation route planning. 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. It is both the fastest and also simplest method.

Analytical and experimental evaluations are provided to validate the usefulness of the proposed approaches. Experimental results show that less than 30% of total edges for contraflow are enough to reduce evacuation time by more than 40% and proposed methods are able to handle about an order of magnitude of larger scenarios than those used in previous works[23, 24]. We also provide the comparison of solution quality between the proposed heuristics and integer programming (optimal contraflow network generator).

1.5 Scope and Outline of the Paper

Our evacuation modeling is based on graph theory with flow analysis. We do not use microscopic simulation that models individual vehicles and measures system optimal performance such as total travel time [14, 22]. Instead, we model traffic load as flow and measure system performance with evacuation time. Our modeling does not include social behavior of evacuees, operational cost 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. Chapter 2 provides a computational framework of the contraflow problem and presents our proposed approaches to solve the contraflow problem. In Chapter 3, we describe design decisions and their analytical evaluations. Chapter 4 presents the experimental setup and evaluation of the approaches. Finally, Chapter 5 summarizes and concludes with a discussion of future work.

Chapter 2

Computational Framework and Proposed Approaches

2.1 Computational Framework

Overload Degree and Overview of Proposed Approaches: In this section, we introduce the computational structure of the contraflow problem using Overload Degree, i.e., the ratio of number of evacuees to bottleneck capacity of transportation network without contraflow, and present appropriate approaches according to each workload zone. As will be shown, the Overload Degree is a key determinant of overall evacuation time, and need for contraflow.

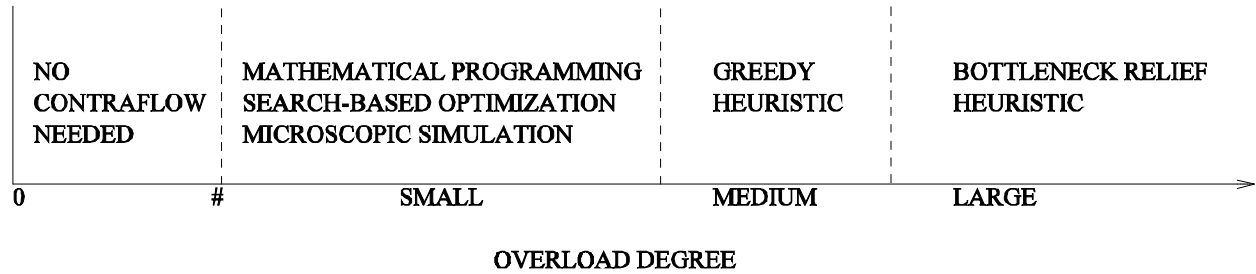


Figure 2.1. Overload Degree and Proposed Approaches

Figure 2.1 illustrates the relationship between Overload Degree and our approaches. In the absence of overload (e.g., Overload Degree is less than one), 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 (e.g., 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 Integer Programming (IP) formulation, explained in Appendix A, 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. We suggest a heuristic based on a greedy approach. Lastly, if the Overload Degree is large, it is close to the case where the network has an infinite source of evacuees. It is necessary to simplify the evacuation modeling to address such a heavy computational workload. We have designed a minimum cut and maximum flow based Bottleneck Relief heuristic that ignores the amount of the population constraint [7].

Use of 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. Flow history of an edge is equivalent to the total number of evacuees who pass through the edge during evacuation time. Figure 2.2 shows how the evacuation route planners interact with our proposed approaches. 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, Greedy heuristic, and Bottleneck Relief heuristic. For the IP approach, the evacuation route planner is combined with a mathematical optimizer to evaluate networks from iterative enumeration and serve as an objective function. The Greedy heuristic uses the flow history of the original network

as input and produces a contraflow reconfigured network. The Bottleneck Relief heuristic uses the original network as input and directly produces a contraflow reconfigured network.

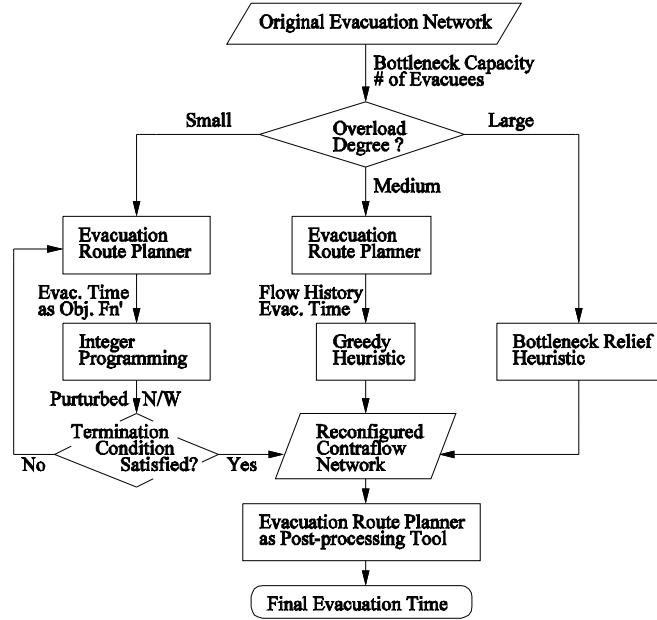


Figure 2.2. Use of Route Planner in the Framework

As network size increases, running the evacuation route planner becomes expensive. Thus, how the evacuation route planner is used is critical in the framework. The Greedy heuristic uses the evacuation route planner one time to generate 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.

2.2 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 congestion history on each edge with the data from the evacuation route planner. We define 'Congestion Index' of an edge e in the following way:

Definition 2. (Congestion Index):

$$\text{CongestionIndex} : (e) = \text{FlowHistory}(e) / (\text{Capacity}(e) \times \text{EvacuationTime})$$

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

The second concept used in the greedy approach is 'Degree of Contraflow'. We can define the Degree of Contraflow in the reconfigured network as follows:

Definition 3 (Degree of Contraflow):

$$\text{Degree of Contraflow (DoC)} = \text{Number of Flipped Edges} / \text{Total Number of Edges}$$

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

Table 2.1. Algorithm Greedy

<p>Algorithm Greedy(G_{original}, DoC);</p> <ol style="list-style-type: none"> 1. run evacuation route planner to produce FlowHistory and Evac.Time on G_{original}; 2. forall edge e in G_{original}, <ul style="list-style-type: none"> CongestionIndex(e) = FlowHistory(e) / (Capacity(e) times Evac.Time); 3. sort edges by CongestionIndex(e) in descending order; 4. $G_{\text{reconfigured}} = G_{\text{original}}$; 5. for each ($i,j$) in the first DoC% edges in the sorted edges, <ul style="list-style-type: none"> $G_{\text{reconfigured}}.\text{flip}((j,i))$; 6. return $G_{\text{reconfigured}}$;
--

The Greedy algorithm shown in Table 2.1 works in the following way. First, we run any evacuation route planner to generate flow history and evacuation time of a given original network. Second, we assign the congestion index value to each edge. Third, the edges are sorted by congestion index in a descending order. Finally, we flip edges in favor of a higher congestion index value among the first DoC% of the sorted edge set. It is required to run the evacuation route planner again over the reconfigured network to get the evacuation time of the reconfigured network.

Example: Figure 2.3 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%. The network in Step 1 is a given original network. If we run an evacuation route planner on the network, we acquire flow history as well as 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 in descending order and the first 60% of them (underlined edges) are greedily selected, as shown in Table 2.2. Each selected edge is compared with its opposite edge and flip opposite edge if the selected edge wins. The final reconfigured network is shown in Step 4, after the flipping process is finished.

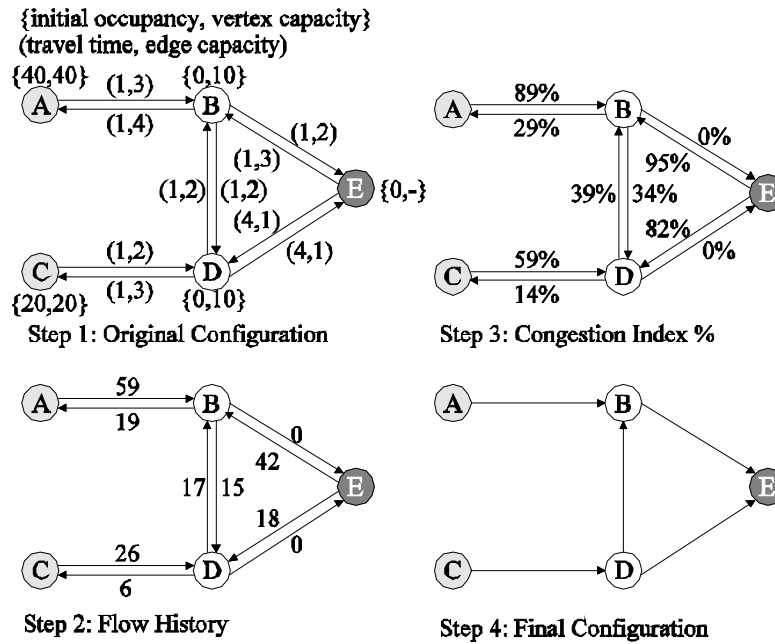


Figure 2.3. Example of Greedy Algorithm

TABLE 2.2. Sorted Congestion Index from Step 3 in Figure 2.3

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

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 flow graph. 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, seen in Step 3 of Figure 2.3.

2.3 Bottleneck Relief Heuristic for Large Overload Degree

The Bottleneck Relief approach starts from the well known theorem, "The value of the max-flow in a capacitated network is equal to the value of min-cut", by Ford and Fulkerson [7]. 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.

TABLE 2.3. Algorithm Bottleneck Relief

<p>Algorithm BottleneckRelief(G);</p> <ol style="list-style-type: none"> 1. while ($\text{maxflow}_{\text{new}} > \text{maxflow}_{\text{old}}$) 2. find mincut of G; 3. flip edges across mincut toward destination; 4. $\text{maxflow}_{\text{old}} = \text{maxflow}_{\text{new}}$; 5. $\text{maxflow}_{\text{new}} = \text{maxflow}(G)$; 6. return G;

If the given graph G has multiple sources and multiple destinations, we have to place a super source connecting to the sources with infinite capacity and a super destination connecting to the destination with infinite capacity before the algorithm BottleneckRelief is applied. The algorithm BottleneckRelief shown in Table 2.3 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 large Overload Degree because maximum flow is based on infinite flow from sources to sinks. Evacuation scenarios over heavily crowded areas as well as 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 finds a minimum cut with high probability in $O(m \log^3 n)$ [15]. In the worst case, our Bottleneck Relief heuristic runs m times, which leads to $O(m^2 \log^3 n)$ runtime.

Example: Figure 2.4 illustrates the application of the Bottleneck Relief heuristic to our simple graph. Nodes A and C are still source nodes while node E is a destination node. The source nodes are connected from a super source 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 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.

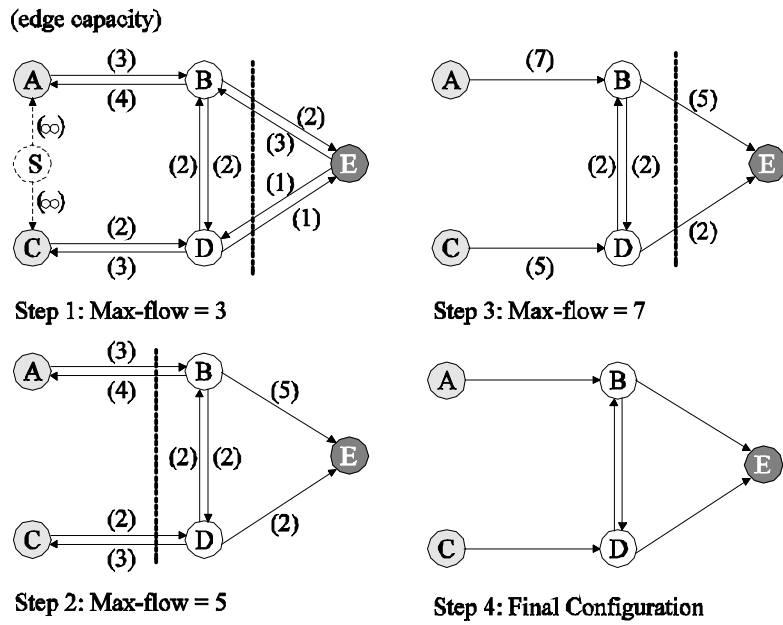


Figure 2.4. Example of Bottleneck Relief Heuristic

Chapter 3

Design Decisions and Their Analytical Evaluations

3.1 Overload Degree and Result Quality

In this section, we explain the relationship between Overload Degree and 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 following the network from this class is minimal. Optimal results are obtained from a huge number of combinatorial network candidates. To achieve an optimal result, we formulate an integer programming (IP) approach (see Appendix A) with an optimal evacuation route planner. The heavy computational load from combinatorial optimization limits the integer programming approach to cases of small Overload Degree as shown in Figure 3.1.

The prohibitive computational workload of achieving optimal results led us to explore effective heuristics. We designed the Greedy heuristic to give priority to more congested edges in the edge flipping mechanism. The heuristic used in this method is simple, but quite effective in identifying critical edges to reduce evacuation time. As detailed in Chapter 4, the result quality of the Greedy heuristic is within the reasonable range from the optimal result. The Bottleneck Relief heuristic also produces solutions of heuristic result quality with fast runtime.

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

Figure 3.1. Dominance Zone of Proposed Approaches

3.2 Choice of Route Planner

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

When we select an evacuation route planner, there is a tradeoff between result quality and runtime. An optimal evacuation route planner can generate optimal evacuation time by performing the following three steps. First, the given network needs to be converted to a time expanded network [11]. Generation of this time expanded network requires knowing the upper bound of the evacuation time. Second, we apply a minimum-cost flow solver to the time expanded network to generate a flow history at each edge. Most minimum-cost flow solvers are based on the linear programming approach. Third, we can extract evacuation time after post-processing the time expanded network with the flow history data. The existing minimum-cost flow solvers (e.g., NETFLO [16], RELAX [2], RNET [10], CS [9], etc.) belong to the optimal

evacuation route planner group. The major drawbacks of optimal evacuation route planners are poor scalability and the requirement of prior knowledge of the upper bound of evacuation time. These linear methods have exponential runtime proportional to a given network size. Thus, they are not appropriate to cover metro size transportation networks, which have easily more than one hundred thousand road segments. The poor estimate of the upper bound of evacuation time leads to either failure of calculation or waste of memory with unnecessary running time.

A heuristic evacuation route planner, by contrast, avoids these issues, often producing close to optimal evacuation time with good scalability. The CCRP (Capacity Constrained Route Planner) algorithm is the only heuristic evacuation route planner available in this domain [19]. 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. The heuristic gives priority to the route with the earliest destination arrival time. Even though the CCRP does not produce optimal solutions in all evacuation scenarios, experiment results show that most solutions are within only 10 percent longer than the optimal evacuation time. In addition, it does not require either preprocessing of a given network or upper bound of evacuation time.

The following subsections present an analytical evaluation of evacuation route planners that illustrates the advantages of our heuristic route planners. We use the following notations to describe the original network: p number of occupancy, n vertices, and m edges.

3.2.1 Optimal Route Planner: RELAX [2]

Bertsekas, who created minimum flow solver RELAX, noted that there is no known polynomial complexity bound for the relaxation method [1]. Thus, we have to depend on the experimental evaluation to measure the performance of RELAX evacuation route planner in combination with the Greedy heuristic.

3.2.2 Optimal Route Planner: Cost Scaling (CS) [9]

The cost-scaling minimum cost flow algorithm combines the ideas of cost-scaling, push-relabel maximum flow method, and the relaxation method. Goldberg incorporated several heuristics (e.g., 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, 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, post-processing 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 $M = (n + m)T + m - \sum_{(i,j) \text{ in } m} \lambda_{ij}$ where λ_{ij} denotes travel time of edge (i,j) [11]. 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 in super-linearly proportion to the number of nodes and evacuees.

3.2.3 Heuristic Route Planner: CCRP [19]

The algebraic cost model of CCRP is presented in [19]. The CCRP evacuation route planner is 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 a 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 maximum edge weight. That is, the combination of the Greedy heuristic with CCRP runs in linearly proportional to the number of nodes and evacuees.

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

Proof: In the Greedy algorithm, step1 runs the evacuation route planner one time and is dominant in terms of runtime. Thus, a direct comparison between optimal and heuristic evacuation route planner runtime 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 heuristic evacuation route planner is faster than that with optimal evacuation route planner when $n^3 p^2 \log(npC) / (m + 2Cn) > 1$ relation holds, which is always true in transportation network.

Lemma 2. The Bottleneck Relief heuristic is faster than the Greedy with CCRP if $p > 9n \log^3 n / (3 + 2C)n$.

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 with CCRP if $p > 9n \log^3 n / (3 + 2C)n$. We can verify the formula using Metropolitan scenario A with a 2 mile zone which has 269,635 (p) occupancy, 562 (n) nodes, and 1443 (m) edges. This dataset fits the sparse network assumption ($m = 3n$). If the parameter values are plugged in, we can observe that the formula satisfies as follows: $(p = 269,635) > (9n \log^3 n / (3 + 2C)n = 21,038)$.

Chapter 4

Experimental Evaluation

4.1 Experiment Setup

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

4.1.1 System Design

We integrated the Integer Programming formulation with CPLEX, a mathematical programming optimizer. CPLEX is a well known commercial optimization tool to solve integer programming, linear programming, and quadratic programming. It includes a sophisticated preprocessor to reduce the size of the search space and a parallel mechanism to speed up the running time. The Greedy heuristic uses an evacuation route planner as a plug-in external module. The two communicate using text files to exchange evacuation time and flow history. This implementation framework gives a flexible structure to accommodate a new evacuation route planner 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. The reconfigured network from the Bottleneck Relief heuristic can also be evaluated by an evacuation route planner.

4.1.2 Dataset Description

Figure 4.1 shows a virtual scenario of a nuclear power plant failure in Monticello, Minnesota. There are twelve 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 radioactive clearance checkup at the designated facility. The demographic data are based on Census 2000 population data. The total number of evacuees is about 42,000. If the given situation is converted to a graph representation, as shown in the figure, the graph has 47 nodes with 148 edges. The graph structure is based on large edge granularity with interstate highway (I-94) and major arterial roads. Thus, the size of the network is relatively small. The interstate highway (path 2,4,15,21,45,29,33,36 and 43) has a larger capacity than other edges. The destination is vertex 40, on the bottom right hand corner of the map. The evacuation time with the original network configuration is 272 min (4 hr 32 min). We created this small size case file (in terms of number of nodes) in order to compare the quality of our heuristics with the optimal network configuration from IP, which has exponential runtime according to network size.

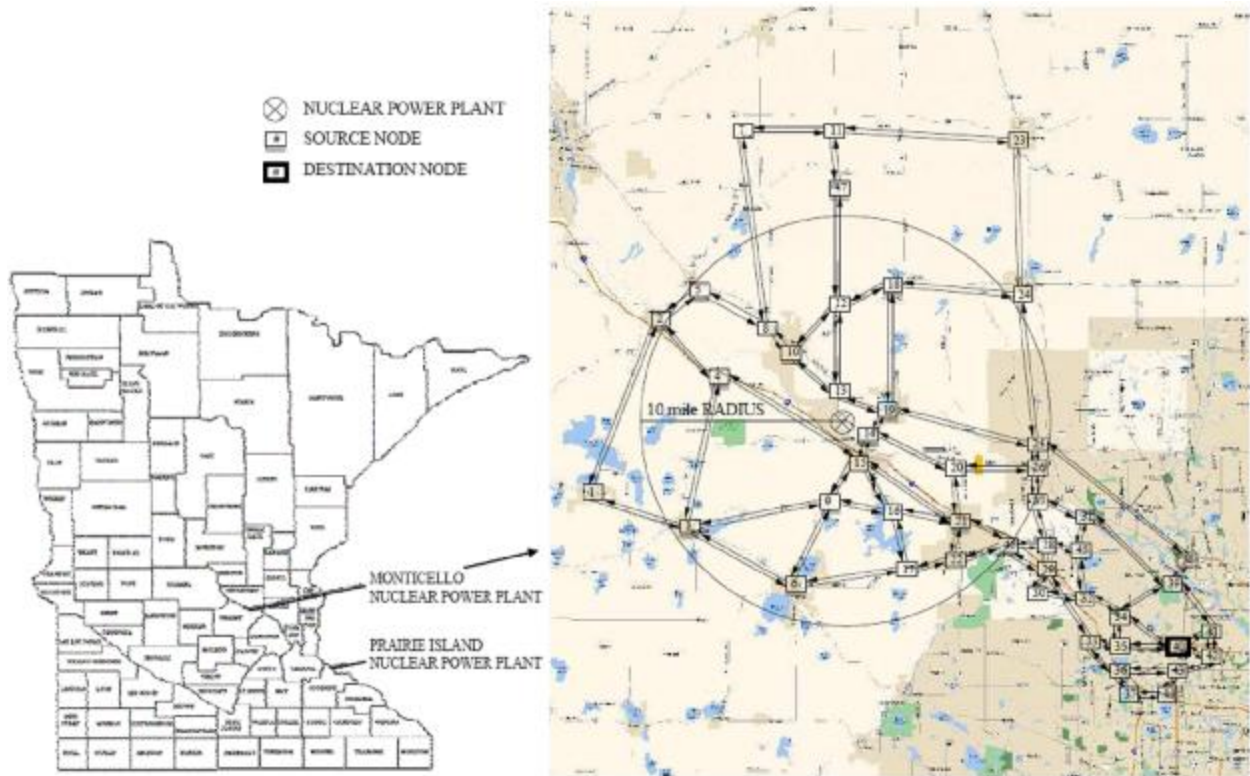


Figure 4.1. Monticello Nuclear Power Plant Case Map

The second dataset was prepared with our evacuation scenario generation software as shown in Figure C-1 in Appendix C. The software has capabilities 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 this software are as follows:

Map Data: consisting of TP+ (planning purpose) and Mn/DOT 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: consisting of Census 2000 data (night time estimation) and Employment data (day time estimation), but not including travelers (e.g., shoppers, etc.).

We selected three different locations as mandated by the Department of Homeland Security with three different network sizes (i.e., 0.5, 1, and 2 mile radius). For security reasons, specific names of locations have been removed in this paper. The primary purpose of the second dataset is to compare the results from heuristic approaches and to test scalability due to the relatively larger network size than the first dataset. The summary of metro evacuation data is described in Table D-1 in Appendix D.

4.2 Overload Degree and Result Quality

As explained previously, Overload Degree is an important parameter to classify the computational structure of the contraflow problem and the proposed heuristics according to the classification of computational workload. In this section, we examine the relationship between Overload Degree and other factors such as evacuation time and runtime of our heuristics using the Monticello dataset. Figure 4.2(a) shows the effects of Overload Degree on evacuation time. We performed this test by changing the number of evacuees over source nodes. We can observe that the evacuation time is linearly proportional to overload degree for all methods. Most heuristics showed reduction of evacuation time by about 30% regardless of Overload Degree. The Greedy heuristic with optimal evacuation route planner (i.e., Relax) always showed minimum evacuation time. The combination of the Greedy with a heuristic evacuation route planner (i.e., CCRP) placed in the middle. The Bottleneck Relief heuristic also showed comparable result quality with the Greedy heuristic.

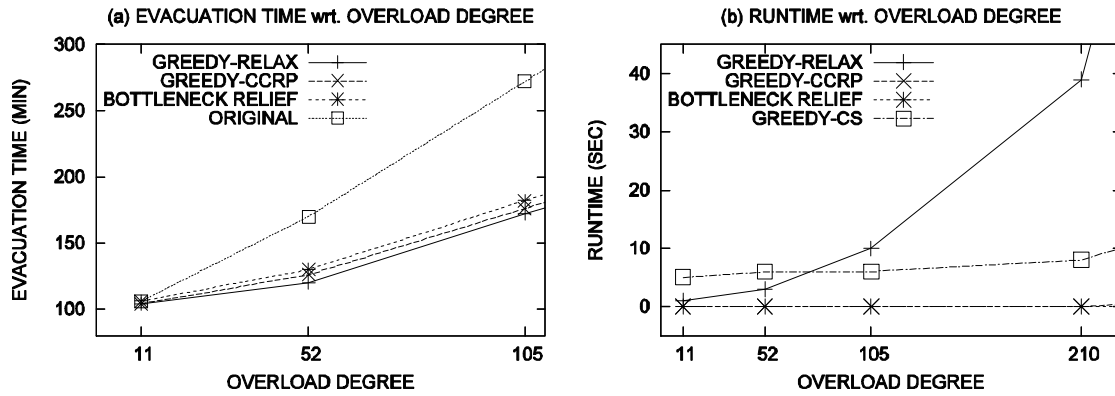


Figure 4.2. Evacuation time / Runtime with regard to Overload Degree using Monticello Scenario

Figure 4.2(b) shows the relationship between Overload Degree and runtime. The Greedy heuristic with RELAX (optimal evacuation route planner) has a steep (maybe, super-linearly) increasing runtime. Greedy with CS (optimal evacuation route planner) showed remarkably faster runtime than Greedy with RELAX as Overload Degree increases. However, Greedy with CCRP (heuristic evacuation route planner) was the fastest combination 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 runtime because it did not involve occupancy data (i.e., number of evacuees) as part of input.

Figure 4.3(a) shows the quality of Greedy heuristics by comparing the results from Integer Programming. First, Greedy heuristics regardless of evacuation route planner showed about 40% reduction of evacuation time. Greedy with optimal evacuation route planner (i.e., RELAX or CS) resulted in only slightly better evacuation time than that with heuristic evacuation route planner (i.e., CCRP). Second, we observe that a gap (i.e., 14 minutes) exists between Greedy heuristics and optimal results. Figure 4.3(b) shows a runtime comparison. Integer Programming resulted in

much higher runtime (i.e., 205 seconds) because the IP formulation took 130,109 iterations to produce optimal contraflow network while Greedy heuristics took only one iteration.

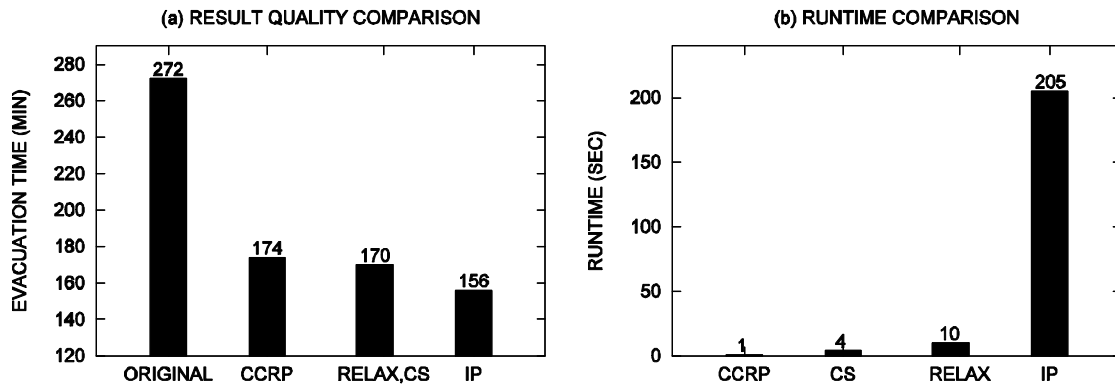


Figure 4.3. Result Quality and Runtime Comparison Between Greedy heuristic and IP Formulation using Monticello Scenario

4.3 Choice of Route Planner and Scalability

Figure 4.4 shows the convergence pattern with regard to Degree of Contraflow using RELAX and CCRP. Although Greedy with RELAX always produced better results than CCRP, the CCRP provided similar result quality with RELAX, showing only a 4 minute gap in evacuation time (RELAX: 170 min, CCRP: 174 min). Both planners also showed similar convergence patterns with regard to Degree of Contraflow. In the Monticello case, less than 10% of total edges contribute to the constantly reduced evacuation time. Figure 13(b) shows a magnified view of evacuation time drop between 10% of Degree of Contraflow. The 10% corresponds to 14 edges out of 147 edges in the Monticello scenario.

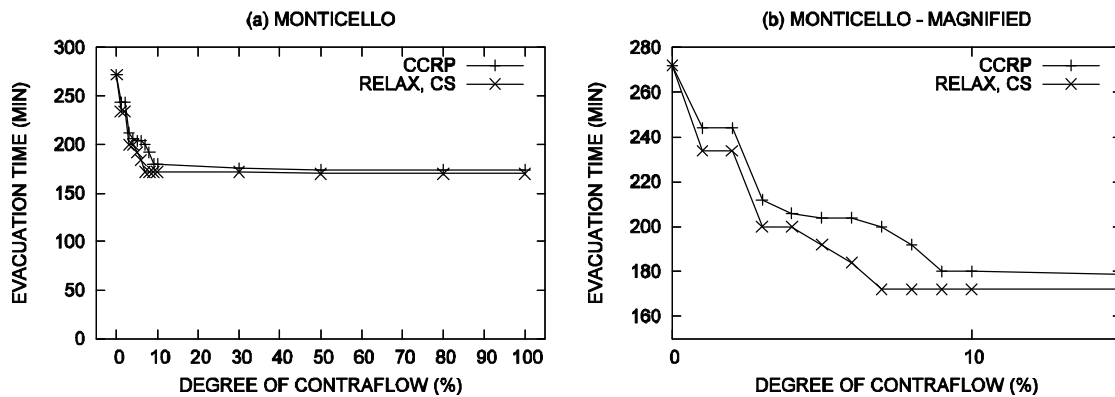


Figure 4.4. Convergence Pattern of Evacuation Time with regard to Degree of Contraflow using Monticello Scenario

Figure 4.5 shows the convergence pattern with metropolitan scenarios. Most evacuation times converged within a 30% Degree of Contraflow. The maximum gap observed between RELAX

and CCRP was 32 minutes (see Figure 4.5(b-1)) and the minimum gap was zero minutes, seen in Figure 4.5(c-1). On average, metro datasets showed a 45% reduction in evacuation time by contraflow from the original to the reconfigured network.

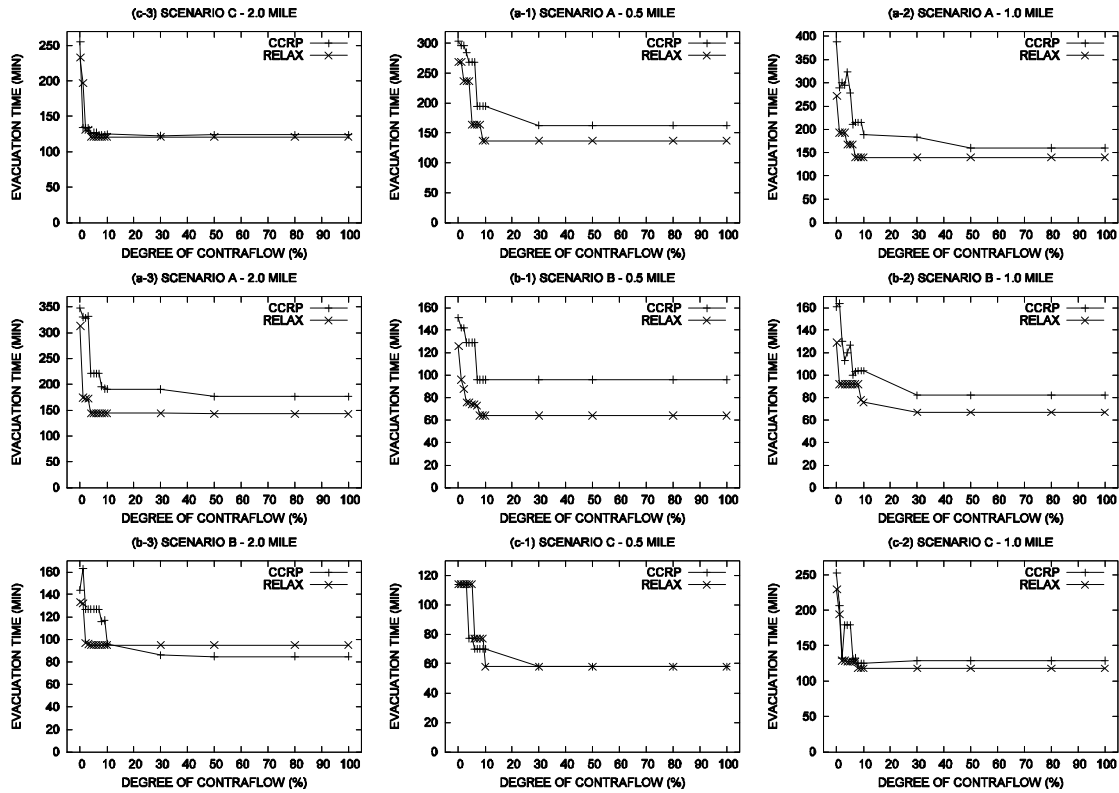


Figure 4.5. Convergence Pattern of Evacuation Time with regard to Degree of Contraflow using Metropolitan Scenario

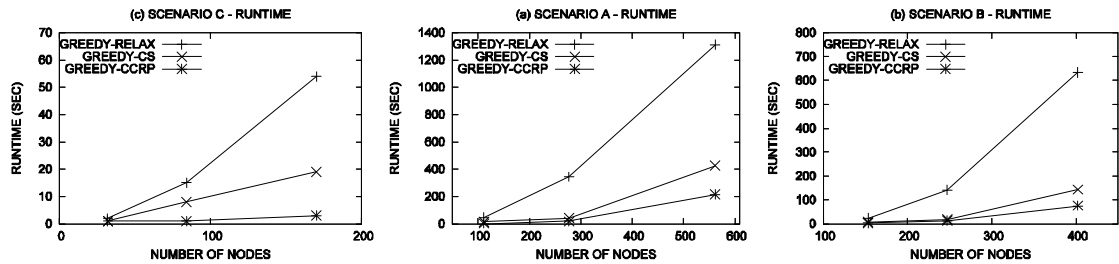


Figure 4.6. Scalability with regard to Network Size using Metropolitan Scenario

Figure 4.6 shows the scalability of the Greedy heuristic with different evacuation route planners using Metropolitan scenarios. The evacuation route planner RELAX showed 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 network size. Nowadays, evacuation at the metropolitan scale is often the issue of interest. In such cases, CCRP will play an important role to scale up our approaches to tackle huge networks.

4.4 Monticello Scenario Results and Implications for Planning

In this section, two findings from the Monticello scenario have especially important implications for evacuation route planning. The first finding is the efficiency of computerized evacuation route planning. Figure 4.7 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 (Figure 16a) results in an evacuation time without contraflow that is twice as long (554 minutes) as that generated by CCRP (276 minutes)(Figure 4.7b). 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.

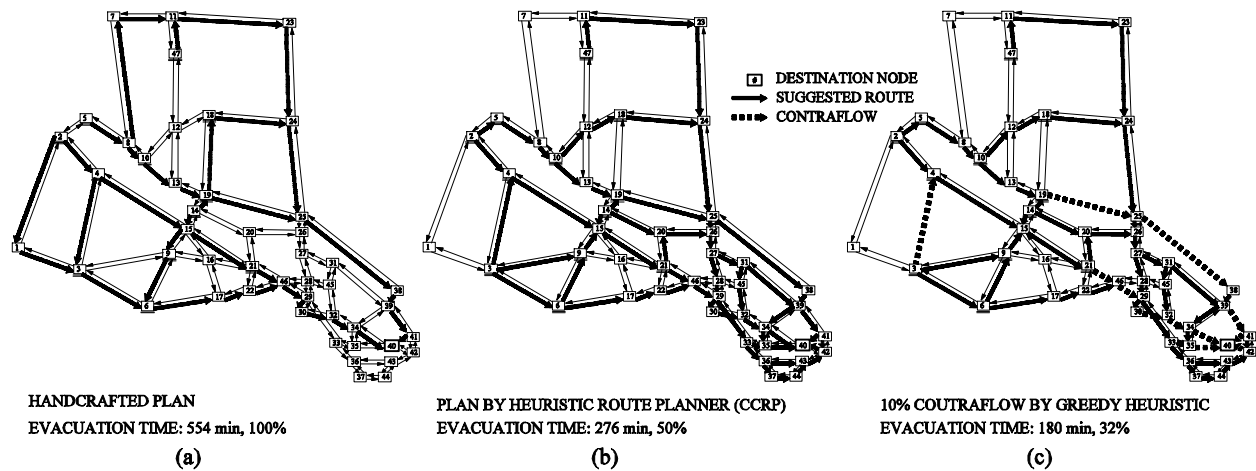


Figure 4.7. Handcrafted vs. Computerized Plans

A second finding is the efficiency of computerized contraflow reconfiguration. On the network shown in Figure 4.7(c), we observe that 10% of 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% of the time required by the original handcrafted version. The value of 10% 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 the major highway with large capacity and local arterial roads around the destination. This selection scheme will help planners to identify and refine more efficient routes for contraflow.

Chapter 5

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 hard problems. Our main contribution was in the fact that we addressed such challenging contraflow problems with computational structure analysis and provided scalable heuristics with high quality solutions. We also presented analytical and experimental evaluations. To summarize the two contraflow heuristics we presented:

Greedy heuristic: guarantees a promising solution quality in spite of its fast run time. Evacuation planning software needs to be interactive due to various combinations of input parameters and evolving datasets. Thus, running time is a critical factor when we implement a computerized contraflow planner. Our well-designed approach based on a Greedy approach 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: is a simplified approach using partial domain knowledge. Its use is suitable for a contraflow situation with large numbers of evacuees. Although we were able to observe comparable result quality with Greedy approach, the runtime of Bottleneck Relief heuristic is fastest regardless of number of evacuees.

Even though 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 evacuation time, which is an essential part of planning.

More in-depth research is required for contraflow algorithms. First, other possible methods should be examined, such as, the possibility of flipping a path instead of an edge. Second, in-bound traffic demand should be considered. Emergency vehicles for traffic officers or fire fighters should have preempted network capacity. Third, partial lane reversal and capacity varying edge need to be incorporated in the modeling.

References

- [1] D. Bertsekas. personal communication by e-mail, available upon request, (2006).
- [2] 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, M.I.T., (Cambridge, MA, 1994).
- [3] R. Caudill and N. Kuo. Development of an interactive planning model for contraflow lane evaluation. *Transportation Research Record, Urban Traffic Systems*, 906(7):47 – 54, (1983).
- [4] J. G. Doheny and J. L. Fraser. Mobedic - a decision modeling tool for emergency situations. *Expert Systems With Applications*, 10:17 – 27, (1996).
- [5] 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 Engineering*, 7:63 – 71, (1992).
- [6] G. Ford, R. Henk, and P. Barricklow. Interstate highway 37 reverse flow analysis - technical memorandum. Technical report, Texas Transportation Institute, (2000).
- [7] L. R. Ford and D. R. Fulkerson. *Flows in Networks*. (Princeton University Press, 1962).
- [8] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP Completeness*. (Freeman, 1979).
- [9] A. V. Goldberg. An efficient implementation of a scaling minimum-cost flow algorithm. *Journal of Algorithms*, 22:1 – 29, (1997).
- [10] M. D. Grigoriadis. An efficient implementation of the network simplex method. *Mathematical Programming Study*, 26:83 – 111, (1986).
- [11] H. W. Hamacher and S. A. Tjandra. Mathematical modeling of evacuation problems: State of the art. *Pedestrian and Evacuation Dynamics*, pages 227 – 266, (2001).
- [12] G. Hamza-Lup, K. A. Hua, M. Le, and R. Peng. Enhancing intelligent transportation systems to improve and support homeland security. In *Proc. of IEEE International Conference on Intelligent Transportation Systems*, (2004).
- [13] T. W. House. Federal response to hurricane katrina lessons learned. <http://www.whitehouse.gov/reports/katrina-lessonslearned/>, (2006).

- [14] M. Jha, K. Moore, and B. Pashaie. Emergency evacuation planning with microscopic traffic simulation. *The Journal of Transportation Research Board*, Transportation Research Record 1886:40 – 48, (2006).
- [15] D. Karger and C. Stein. An $O(n^2)$ algorithm for minimum cuts. In *Proceedings of the 25th ACM Symposium on the Theory of Computing*, pages 757 – 765, (1993).
- [16] J. L. Kennington and R. V. Helgason. *Algorithm for Network Programming*. (Wiley and Sons, 1980).
- [17] S. Kim and S. Shekhar. Contraflow network reconfiguration for evacuation planning: A summary of results. In *Proceedings of the 13th ACM Symposium on Advances in Geographic Information Systems*, pages 250 – 259, (2005).
- [18] T. Litman. Lessons from katrina and rita: What major disasters can teach transportation planners. *Journal of Transportation Engineering*, 132(1):11 – 18, (2006).
- [19] Q. Lu, B. George, and S. Shekhar. Capacity constrained routing algorithms for evacuation planning: A summary of results. *Proceedings of 9th International Symposium on Spatial and Temporal Databases*, pages 291 – 307, (2005).
- [20] M. W. C. of Governments. 1998 performance of regional high-occupancy vehicle facilities on freeways in the Washington region. *Analysis of Person and Vehicle Volumes*, (1999).
- [21] G. Theodoulou. Contraflow evacuation on the westbound i-10 out of the city of new orleans. Master's thesis, Louisiana State University, (2003).
- [22] G. Theodoulou and B. Wolshon. Alternative methods to increase the effectiveness of freeway contraflow evacuation. (*The Journal of Transportation Research Board*, Transportation Research Record 1865:48 – 56, (2004).
- [23] H. Tuydes and A. Ziliaskopoulos. Network re-design to optimize evacuation contraflow. Technical Report 04-4715, Presented at 83rd Annual Meeting of the Transportation Research Board, (2004).
- [24] H. Tuydes and A. Ziliaskopoulos. Tabu-based heuristic for optimization of network evacuation contraflow. Technical Report 06-2022, Presented at 85th Annual Meeting of the Transportation Research Board, (2006).
- [25] B. Wolshon. One-way-out: Contraflow freeway operation for hurricane evacuation. *Natural Hazards Review*, 2(3):105 – 112, (2001).
- [26] B. Wolshon, E. Urbina, and M. Levitan. National review of hurricane evacuation plans and policies. Technical report, Hurricane Center, Louisiana State University, Baton Rouge, Louisiana, (2002).

Appendix A
Optimal Contraflow Solution using Integer Programming

Before the complete IP formulation is presented, we explain the constants and variables used in our formulation in Table A-1.

TABLE A-1. Variables used in Contraflow Integer Programming Formulation

N	set of Nodes
N_s, N_k	set of source and sink nodes
E	set of edges
T	predetermined upper bound of total travel time
IO_i	initial occupancy in node i , for i in N
VC_i	vertex capacity of node i , for i in N
$EC_{(i,j)}$	edge capacity of edge (i,j) , for (i,j) in E
$ET_{(i,j)}$	travel time of edge (i,j) , for (i,j) in E
$f_{(i,j),t}$	amount of flow on edge (i,j) during interval $(t, t + ET_{(i,j)}]$, for (i,j) in E , $0 \leq t \leq T$
$g_{(i,j),t}$	amount of contraflow on edge (j,i) during interval $(t, t + ET_{(j,i)}]$, for (j,i) in E , $0 \leq t \leq T$
$n_{i,t}$	number of occupancy which remain in node i during interval $(t, t + 1]$, for i in N , $0 \leq t \leq T$
F_t	1 iff there is flow on any edge at interval $(t-1, t]$, for $1 \leq t \leq T$
$cf_{(i,j)}$	1 iff edge (i,j) is used for the contraflow, (0 o.w.), for (i,j) in E

$g_{(i,j),t}$ happens only when an edge (j,i) is reversed, i.e. (i,j) is selected for the contraflow. $n_{i,t}$ can be considered as a flow $f_{(i,j),t}$ which flows in terms of time from the node i to itself, but for simplicity of notation, $n_{i,t}$ is used.

Figure A-1 depicts the relationship between flow variables (including $n_{i,t}$). Fix a node (i) and the time slot (t) and consider all the flow variables for the node i around the time slot t . Then the flow variables can be categorized into four different kinds as follows:

a is a variable for flow from a node other than i that reaches the node i at the time slot t . $f_{(j,i),t-ET_{(j,i)}}$ and $g_{(j,i),t-ET_{(j,i)}}$ are flow variables of type a.

b is a variable for flow from i to i during the time interval $(t-1, t]$. $n_{i,t-1}$ is a flow var. of type b.

c is a variable for flow from i to i during the time interval $(t, t+1]$. $n_{i,t}$ is a flow var. of type c.

d is a variable for flow to a node other than i that leaves the node i at the time slot t . $f_{(i,j),t}$ and $g_{(i,j),t}$ are flow variables of type d.

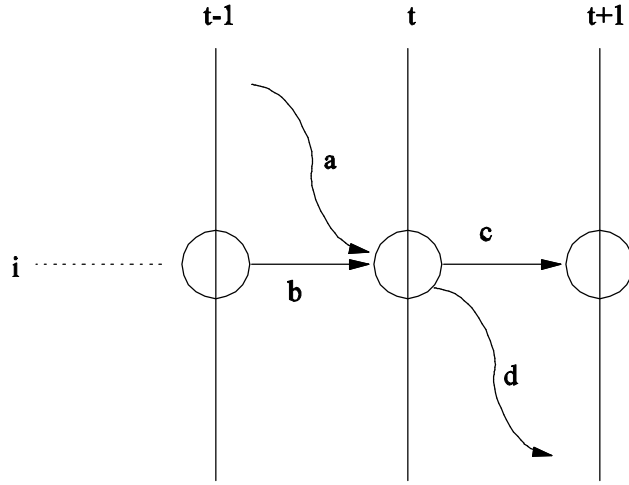


Figure A-1. Flow variables in MIP formulation

Since we are dealing with time-expanded graph, we need flow variables of type b and c as well as a and d to represent the flow which stays at a node temporarily. Note that when $t=0$, the flow of type a is the initial occupancy of the node and there is no flow of type b.s

Equation (1) in Table A-1 defines the objective function such that $\sum_{t=1}^T F_t$ is the total amount of time to finish the evacuation assuming T is large enough. If T is set to less than the minimum value, then the formulation becomes infeasible. (2) is used to set $F_t = 1$ if there is flow on any edge at the time slot t . (3) and (4) describe the initial and general flow conservation constraints, respectively. What it means is $a + b$ equals $c + d$, i.e. inflow equals outflow. (5) is the evacuation termination constraint which means on or before the time slot of T , the evacuation must be terminated. Note that the equation in (5) means at time of t , the destination nodes have all the evacuees and from that time on, no outflow will happen except $n_{i, k}$, for i in $N_{k, k} \geq t$. (6) is a constraint to make the flow happen successively. (7) 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; normal flows or one contraflow. We do not consider the case of reversing the two edges at the same time. Finally, (8) is used to ensure the proper allowed amount of flow on each edge based on the value of the contraflow variable.

The above formulation depends on the value of T which should be at least as large as the minimum evacuation time. If T is set too big, it will increase the complexity of the formulation and hence the computation time will become unreasonable. While if T is set too small, the formulation will become infeasible. In order to solve the problem using the formulation, we have two possible options: decreasing T or increasing T . Since it's easy to compute the minimum evacuation time using only normal flows, we can set T to be the computed time and run IP solver. Or if the IP solver can find infeasibility very efficiently for T which is less than the minimum value, then we can think of starting from a small T and increasing T until we find a feasible solution. Our experimental results show that the latter approach, increasing T , is quite efficient.

Further analysis of the efficient solution techniques in theoretical and experimental points of view is necessary and lies in the next step of this work. However in this work, we include the optimal solutions to compare those with results from our heuristics.

TABLE A-2. Contraflow Integer Programming Formulation

#	Equation	Description
(1)	$\min \sum_{t=1}^T F_t$	objective function (i.e. minimize evacuation time)
(2)	$\sum_{\substack{(i,j) \in E \\ t-ET_{(i,j)} \leq k \leq t-1}} [f_{(i,j),k} + g_{(j,i),k}] \leq \sum_{\substack{(i,j) \in E \\ t-ET_{(i,j)} \leq k \leq t-1}} EC_{(i,j)} F_t,$ $t = 1, \dots, T$	Setting the existence of flow on edge constraint
(3)	$n_{i,0} = IO_i - \sum_{(i,j) \in E} f_{(i,j),0} - \sum_{(j,i) \in E} g_{(j,i),0}, \forall i \in N$	Initial flow conservation constraint
(4)	$n_{i,t} = n_{i,t-1} + \sum_{(j,i) \in E} f_{(j,i),t-ET(j,i)} + \sum_{(i,j) \in E} g_{(j,i),t-ET(i,j)}$ $- \sum_{(i,j) \in E} f_{(i,j),t} - \sum_{(j,i) \in E} g_{(j,i),t}, \quad \forall i \in N, t = 1, \dots, T$	General flow conservation constraint
(5)	$\sum_{i \in N_k} n_{i,T} = \sum_{i \in N_s} IO_i$	Evacuation termination constraint
(6)	$\sum_{k=t+1}^T F_k \leq (T-t)F_t, \quad t = 1, \dots, T$	Successive flow constraint
(7)	$cf_{(i,j)} + cf_{(j,i)} \leq 1, \quad \forall i, j \text{ s.t. } (i,j), (j,i) \in E$ $cf_{(i,j)} \leq 1, \quad \forall i, j \text{ s.t. } (i,j) \in E, (j,i) \notin E$ $cf_{(j,i)} \leq 1, \quad \forall i, j \text{ s.t. } (j,i) \in E, (i,j) \notin E$	Contraflow flipping constraint
(8)	$f_{(i,j),t} \leq EC_{(i,j)}(1 - cf_{(i,j)}), \quad \forall (i,j) \in E, t = 0, \dots, T$ $g_{(j,i),t} \leq EC_{(j,i)}cf_{(j,i)}, \quad \forall (j,i) \in E, t = 0, \dots, T$	Contraflow capacity constraint
(9)	$F_t \in \{0,1\}, \quad t = 1, \dots, T$	F_t is either 0 or 1
(10)	$0 \leq f_{(i,j),t} \text{ and } f_{(i,j),t} \in \mathbf{Z}, \quad \forall (i,j) \in E, t = 0, \dots, T$	Integer constraint
(11)	$0 \leq g_{(i,j),t} \text{ and } g_{(i,j),t} \in \mathbf{Z}, \quad \forall (i,j) \in E, t = 0, \dots, T$	Integer constraint
(12)	$0 \leq n_{i,t} \leq VC_i \text{ and } n_{i,j} \in \mathbf{Z}, \quad \forall i \in N, t = 0, \dots, T$	Node capacity constraint
(13)	$cf_{(i,j)} \in \{0,1\}, \quad \forall (i,j) \in E$	$cf_{(i,j)}$ is either 0 or 1

Appendix B
How Hard Is Contraflow Problem?

As of editing time of this paper, we conjecture that the contraflow problem is NP complete. The sketch of proof outline is described in this section. In general, the process of devising an NP completeness proof for a decision problem Π consists of the following four steps [8].

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.

In our process of proof, we select the known NP complete problem as 3-SATISFIABILITY (3SAT) problem which is almost the root of other NP complete problems and is derived from SATISFIABILITY problem whose NP completeness is proven by Cook [8]. The 3SAT problem is specified as follows:

3SAT

INSTANCE: Collection $C = \{c_1, c_2, \dots, c_m\}$ of clauses on a finite set U of variables such that $|c_i| = 3$ for $1 \leq i \leq m$.

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 evacuation time of a given graph. For simplicity, each edge in an undirected graph G should be flipped in either way.

CONTRAFLOW

INSTANCE: An undirected graph $G = (V, E)$ with initial occupancy $o(v) \in Z^+$ (where Z^+ denotes the positive integers) for some $v \in V$, destination vertices for some $v \in V$, capacity $c(e) \in Z^+$ and travel time $t(e) \in Z^+$ for each $e \in E$, a directed graph $G' = (V, E')$ and evacuation time bound $B \in Z^+$.

QUESTION: Is there a function $f: e \rightarrow \{u, v\}, \{v, u\}$ for each $e \in E$ where $\{u, v\}$ or $\{v, u\}$ is a directed edge in E' such that $EVAC-TIME(G') \leq B$?

Conjecture 1. CONTRAFLOW is NP complete.

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

We transform 3SAT to CONTRAFLOW. Let $U = \{u_1, u_2, \dots, u_n\}$ and $C = c_1, c_2, \dots, c_m$ be any instance of 3SAT. We must construct a graph $G' = (V, E')$ and a positive integer B such that G' has an 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 $\{s_1, s_2, \dots, s_m\}$ with $o(s) = 1$. The destination component consists of two layers. First layer consists of each literals and their negated literals in U (i.e., $u_1, \bar{u}_1, u_2, \bar{u}_2, \dots, u_n, \bar{u}_n$). Second layer consists of XOR of each pair of literals (i.e., $u_1 \oplus \bar{u}_1, u_2 \oplus \bar{u}_2, \dots, u_n \oplus \bar{u}_n$). This XOR layer serves as a destination node set in the CONTRAFLOW problem. The two nodes in a pair (u_i and \bar{u}_i) in the first layer are connected to each XOR node ($u_i \oplus \bar{u}_i$) in the second layer with edges each of whose $t(e) = 1$ and $c(e) = 1$. Finally, a flipping component consists of edges with the following definition. For each clause $c_j \in C$, let the three literals in c_j be denoted by x_j, y_j , and z_j . Then, the edges are $\{s_j, x_j\}, \{s_j, y_j\}, \{s_j, z_j\}$ each of whose $t(e)=0$ and $c(e)=1$. Figure 18 shows an example of the contraflow graph obtained when $U = \{u_1, u_2, u_3, u_4\}$ and $C = \{\{u_1, \bar{u}_3, \bar{u}_4\}, \{\bar{u}_1, u_2, \bar{u}_4\}\}$.

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 $EVAC-TIME(G') \leq B$ by flipping edges in G to prove that the above construction is indeed a transformation.

à: Suppose that C is satisfiable. If we define the function f as $e = \{u, v\}$ if v is TRUE or $e = \{v, u\}$ if v is FALSE (i.e., draw arrow head on the TRUE node and 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 occupancy 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 destination component.

β: Suppose that $EVAC-TIME(G') \leq B$ by using the same flipping function f described above. For each occupancy 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 destination component. This guarantees that C is satisfiable.

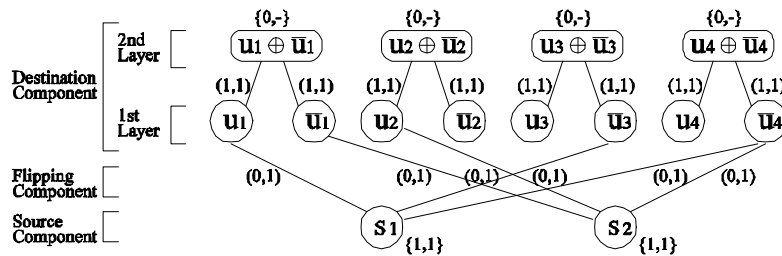


Figure B-1. CONTRAFLOW instance resulting from 3SAT instance in which $U = \{u_1, u_2, u_3, u_4\}$ and $C = \{\{u_1, \bar{u}_3, \bar{u}_4\}, \{\bar{u}_1, u_2, \bar{u}_4\}\}$.

Appendix C
Evacuation Scenario Generation Software

The following screenshot shows the interface of evacuation scenario generation software.

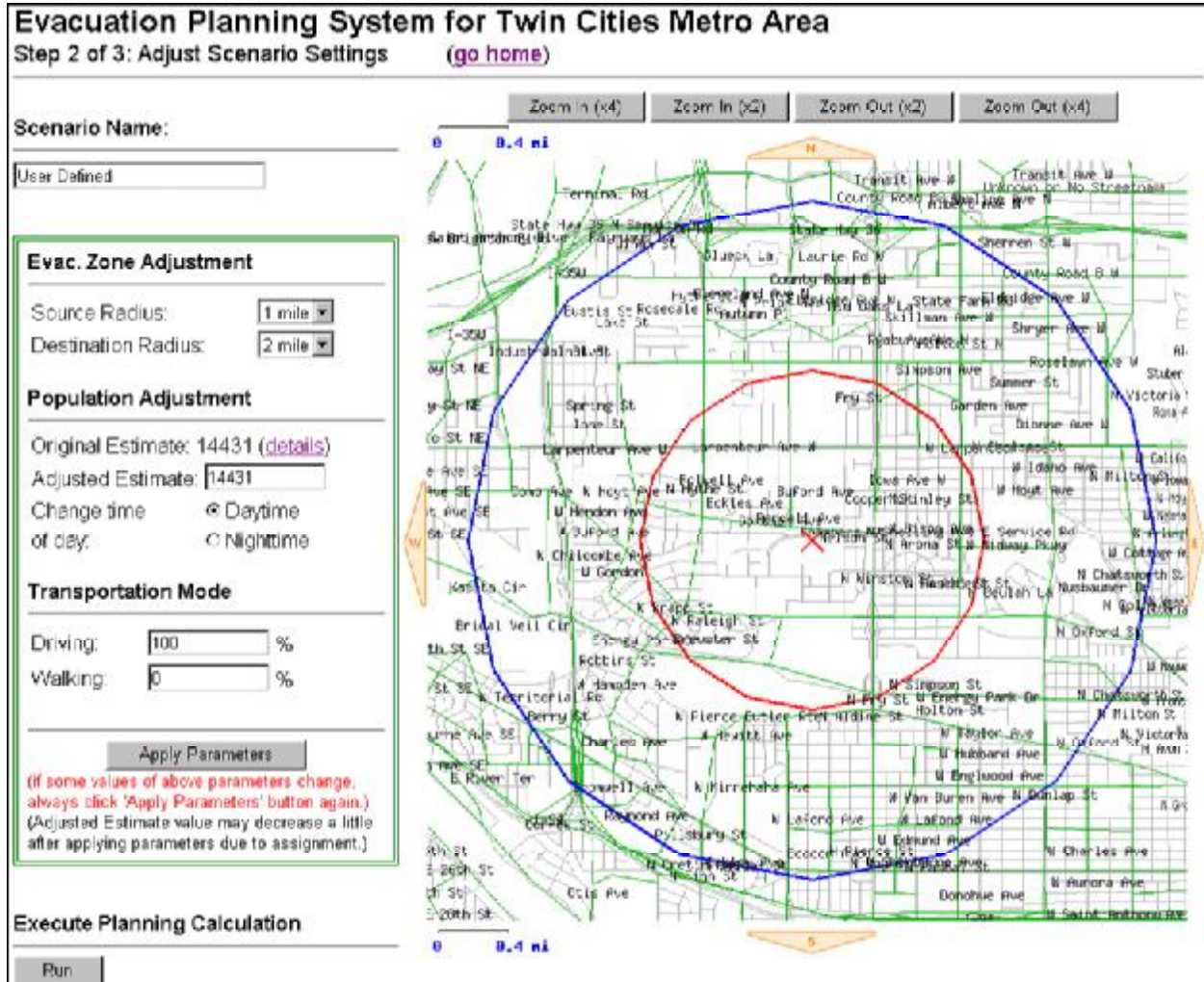


Figure C-1. Screenshot of Evacuation Scenario Generation Software

Appendix D
Summary of Metro Evacuation Dataset

Scenario names are removed for security reason.

Table D-1.

Scn.	Size	Occ.	Node/Edge	Scn.	Size	Occ.	Node/Edge	Scn.	Size	Occ.	Node/Edge
A	0.5 mi	117,643	111 / 287	B	0.5 mi	53,938	153 / 369	C	0.5 mi	8,878	32 / 55
	1.0 mi	148,007	277 / 674		1.0 mi	84,678	247 / 608		1.0 mi	27,406	84 / 159
	2.0 mi	269,635	562 / 1143		2.0 mi	139,994	402 / 1033		2.0 mi	43,689	170 / 381

Appendix E
How to Measure Evacuation Time?

The evacuation time used in our model is the earliest destination arrival time (i.e., earliest exit time): the period from the moment when the first evacuee leaves his/her source node to the moment when the last evacuee arrives certain destination node. To understand the details of measuring the evacuation time, it is necessary to know two important concepts in graph theory: time expanded graph and minimum cost flow theory.

We can consider the time expanded graph as a discrete time expansion of a static network flow problem [11]. Given a directed network $G(V, E)$, we can define the time expanded graph G_T in the following way.

Definition 4. (Time Expanded Graph): Let $G(V, E)$ be a directed network with V the set of nodes with initial occupancy V_{occ} and node capacity V_{cap} and E the set of edges with travel time $\lambda_{id1,id2}$ and edge capacity. The time expanded graph $G_T(V_T, E_T)$ associated with $G(V, E)$ over a time horizon T is defined as:

S is super source and D is super destination.

$$V_T := \{V_{id,t} \mid id \in \text{node } id \text{ of } V \text{ and } t = 0, 1, \dots, T\}$$

$$E_T := \{(V_{id,t}, V_{id,t+1}) \mid id \in \text{node } id \text{ of } V \text{ and } t = 0, \dots, T-1\}$$

$$\cup \{(V_{id1,t1}, V_{id2,t2}) \mid id1, id2 \in \text{node } id \text{ of } E_{id1,id2} \text{ and } t2 = t1 + \lambda_{id1,id2}\}$$

$$\cup \{(S, V_{id,0}) \mid id \in \text{node } id \text{ of } V \text{ with } V_{occ} \} > 0\}$$

$$\cup \{(V_{id,t}, D) \mid id \in \text{node } id \text{ of } V \text{ with } V_{occ} = \infty \text{ and } t = 1, \dots, T\}$$

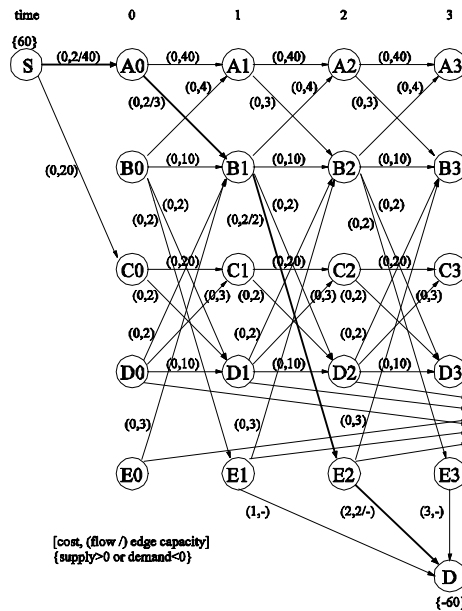


Figure E-1. Time Expanded Version from the Graph in Figure 3(a)

In the above definition, the $(V_{id,t}, V_{id,t+1})$ edge is called a holdover edge because when a flow runs through $V_{id,t}$ and $V_{id,t+1}$ it means that evacuees corresponding to the flow size are staying in the same node $V_{id,t}$ during the time period t and $t+1$. On the other hand, the $(V_{id1,t1}, V_{id2,t2})$ edge is

called a movement edge because evacuees corresponding to the size of flow running through the two nodes are moving from V_{id1} to V_{id2} in original network G . In Figure E-1, edge A0-A1 is an example of a holdover edge while edge A0-B1 is an example of a movement edge.

The time horizon T should be greater than the final evacuation time. Thus we need to set an arbitrary great value to T when we run a case first time. After the first run, we should reduce T as close to the evacuation time as possible to reduce system memory and run time. The unit time should not necessarily be 1 second. The choice of unit time depends on the model realism and complexity.

It would be easy to understand if we follow the steps of constructing the time expanded graph shown in Figure E-1 derived from Figure 1.3(a). The graph in Figure 1.3(a) will be called 'original graph' in this description. Let us start with showing how nodes are generated in time expanded graph. Each column nodes (e.g., A0, B0, ..., E0) in the time expanded graph in Figure E-1 correspond to the original graph at the given time (e.g., time 0). Thus each node id in time expanded graph (e.g., 'A0') represents the combination of node id in original graph (e.g., 'A') and time (e.g., time '0'). We duplicate the original graph over time period from 0 to T (time horizon). At first, the value T should be set large enough to exceed evacuation time. If we get approximate evacuation time later, the T should reduce to slightly greater than the evacuation time to save memory and reduce running time. In the next step we add super source(id 'S') with supply corresponding to the total number of evacuees in original graph (e.g., $40 + 20 = 60$) and super destination(id 'D') with demand of the same value of supply in negative. Now we generated all the necessary nodes in a time expanded graph.

Edges are constructed in the following way. Edges from super source 'S' go to the nodes with initial occupancy at time 0 (e.g., A0 and C0). Edges to super destination 'D' come from the destinations in original graph at time > 0 . Edges connecting two nodes with same node id are holdover edges. If flow goes through the holdover edge, it corresponds to evacuees staying from time t to time $t+1$ (e.g., flow from A0 to A1 means that evacuees stay at node A from time 0 to 1). Edges connecting two nodes with different node ids are movement edges. If flow goes through the movement edge, it corresponds to evacuees running between the nodes (e.g., flow from A0 to B1 means that evacuees run from A to B from time 0 to 1). Especially, all the holdover edges and movement edges have 0 travel time. Only edges between destination nodes and super destination D have travel time corresponding to their time. This special travel time assignment is the key point of a time expanded graph and enables minimum cost flow theory.

After constructing the graph structure, we send flow from S to D through time expanded graph. One example flow is the thick edges on the figure. The maximum capacity along the path is two. Thus, only two evacuees can escape through the path. The travel times of edges (A,B) and (B,E) in the original graph are respectively 1. Thus, the two evacuees should go through node E2 on time expanded graph because it takes two time units in the original graph. If we find a set of shortest paths based on available capacity, the set naturally becomes the evacuation plan because each flow amount in the flow set corresponds to a group of evacuees escaping in minimal evacuation time. The solution produced by minimum cost flow algorithm is always optimal in the sense that we can acquire the minimum evacuation time.