Contraflow Network Reconfiguration for Evacuation Planning: A Summary of Results

Sangho Kim
Department of Computer Science
University of Minnesota
sangho@cs.umn.edu

Shashi Shekhar
Department of Computer Science
University of Minnesota
shekhar@cs.umn.edu

ABSTRACT

Contraflow, or lane reversal, is a way of increasing outbound capacity of a real network by reversing the direction of inbound roads during evacuations. The contraflow is considered a potential remedy to solve congestions during evacuations in context of homeland security and natural disasters (e.g., hurricanes). Currently available contraflow algorithms only tackle a single-source and multiple-destinations situation. These approaches cannot handle a multiple-sources problem which is harder due to conflicts across the optimal paths from different sources.

We formally define the evacuation situations using graph and flow theory and show the NP completeness of the contraflow problem. We propose two capacity-aware global contraflow heuristics that produce contraflow configuration in the presence of conflicts among routes preferred by different source nodes. We evaluate proposed heuristics experimentally using synthetic networks as well as real world datasets. In addition, we provide algebraic cost model. Experimental results show that our contraflow heuristics can reduce evacuation time by 30% or more.

Categories and Subject Descriptors

F.2.0 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity—*General*

General Terms

Algorithms, Experimentation

Keywords

contraflow, lane reversal, evacuation planning, time expanded graph, combinatorial optimization, simulated annealing

1. INTRODUCTION

Evacuation planning is currently an issue of major importance due to the increasing risks both from terrorist attacks and natural disasters. From the perspective of transportation system, colossal traffic jams during evacuation process have been the main issue. In the aftermath of hurricanes George in 1998 and Floyd in 1999, the transportation community observed the need for increased evacuation route capacity as well as more accurate estimate of evacuation time[22].

Contraflow, or lane reversals, has been discussed as a potential remedy to solve such tremendous congestion by increasing outbound evacuation route capacity. Today, eleven of the 18 coastal states threatened by hurricanes consider the use of contraflow as part of their evacuation strategy[21]. Although contraflow is primarily important for evacuations, its applications are not limited. 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 peak time. The utilization of contraflow after football games is another typical example of a source, multiple destinations situation.

In spite of its importance and various applications, researchers have yet to address the contraflow problem from a computational perspective. Computerized contraflow design may help not only developing the optimized network configuration but also quantifying the evacuation time of the result network.

Thus, we formulate the contraflow problem with the help of graph and flow theory. With a given evacuation situation using a directed and capacitated graph with multiple sources and multiple destinations, we want to find a reconfigured network by contraflow with the objective of minimizing evacuation time. In addition to the problem formulation, we conjecture that the contraflow problem is NP complete and provide proof outline. The understanding of NP completeness on the problem is a stepping stone to the application of combinatorial optimization techniques to the contraflow problem.

To our knowledge, algorithms described in [14] are the only ones tackling the contraflow problem. In their evacuation modeling, an evacuation zone consists of a source and multiple destinations. Their solution is reduced to finding the optimal paths from the single source to the destinations and overlaying them. This approach is not effective when there are one or more specific destinations for evacuees, who are located at multiple nodes in the transportation network. The approach breaks down due to conflicts across the optimal paths from different sources.

On the other hand, we present capacity-aware global contraflow planning heuristics that can produce contraflow net-

GIS'05, November 4, 2005, Bremen, Germany.

work configuration in the presence of multiple sources with conflicting paths to destinations. Our first approach is Flip High Flow Edge (FHFE) which flips edges with priorities and produces an appropriate level of result quality with fast running time. Our second approach is based on Simulated Annealing (SA) which aggressively improves evacuation time by escaping local minima in an iterative way. Each approach is evaluated and examined with both synthetic transportation networks for scalability test as well as a nuclear power plant failure scenario in Monticello, Minnesota, U.S.. Both real and synthetic datasets show that the evacuation time of our FHFE is close to that of the SA in spite of its fast running time. In addition, our heuristics can reduce evacuation time by a third.

The rest of the paper is organized as follows. Section 2 formulates the contraflow problem and shows NP completeness of the contraflow problem. Section 3 presents related work. In Section 4, we describe the two heuristics. Section 5 presents experiment setup information and evaluation of the approaches. Section 6 discusses related knowledge to understand the contraflow problem. Finally, section 7 summarizes and concludes with a discussion about future work.

2. PROBLEM DEFINITION

Section 2 consists of two subsections. The first subsection formulates the contraflow problem and gives an illustrative example. The second subsection discusses the NP completeness of the contraflow problem.

2.1 Problem Formulation

We understand the evacuation 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(V, E) be a directed network with V, the set of vertices and E, the set of edges. Each vertex has an initial occupancy value, that is, the number of residents to evacuate and vertex capacity. Each edge also has an edge capacity and constant travel time. The evacuation situation can have multiple source vertices and destination vertices. Evacuation time is defined as a period from the moment when a first evacuee leaves a source vertex to the moment when a last evacuee arrives at a destination vertex. We want to find a reconfigured network by contraflow with the objective of minimizing evacuation time. The following is the summary of our contraflow problem.

Given: 1. Transportation network, directed graph G(V,E)
2. Each vertex has initial occupancy and capacity.
3. Each edge has capacity and travel time.
4. Source and destination vertices.

Find: Contraflow network configuration

Objective: Minimize evacuation time.

Constraint:
1. Travel time and capacity are constant.
2. Flipping a portion of the lanes is not allowed.

Figure 1 illustrates a simple evacuation situation on a transportation network. Each vertex represents a city with initial occupancy and its capacity. City A has 40 people and also capacity 40. Vertices A and C are modeled as source vertices, while vertex E is modeled as a destination vertex (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 maximum 3 evacuees can simultaneously travel through the edge. Vertices B and D have no initial occupancy and only work as transshipment vertices. If we apply a minimum cost flow algorithm to this configuration, evacuation time is 22.

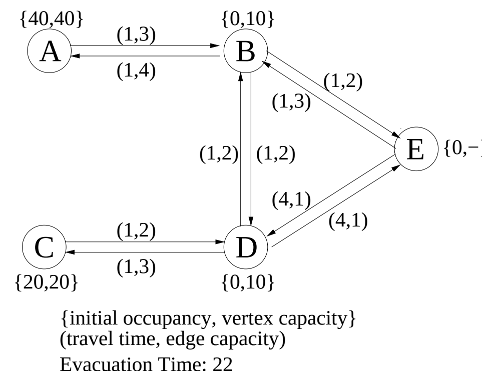

Figure 1: Graph Representation of an Evacuation Situation

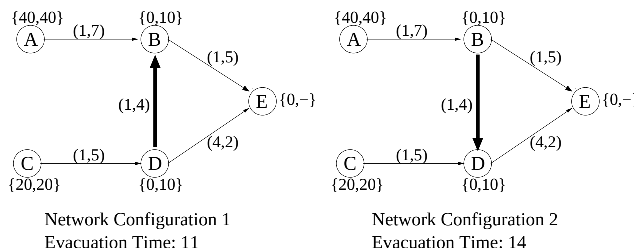

Figure 2: Two Possible Contraflow Configurations from Figure 1

Figure 2 illustrates two possible contraflow configurations based on the graph in Figure 1. 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 vertices B and D. If we apply the minimum cost flow algorithm to both configurations, the left configuration has evacuation time 11, while the right configuration 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 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 2.

In addition, we can identify some drawbacks of existing approach with this example. Suppose that the graph in Figure 1 is solved by overlaying Dijkstra's shortest paths. One path is A, B, and E and the other path is C, D, B and E. Thus, the resulting paths do not utilize all the edges effectively because the solution does not consider flow and capacity of the given graph. If we assume that this example is a multiple-sources and multiple-destinations (e.g., E', E") problem, it is possible that two paths, A-B-D-E' and C-D-

B-E", may conflict on the edge (B, D).

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. Incorporating microscopic traffic models into our heuristics has tradeoff between performance and model realism. Second, we only consider reversing whole lanes for simplicity. In an actual implementation of contraflow, it is possible to reverse some portion of lanes.

2.2 How Hard Is The Contraflow Problem?

As of the 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 = $c_1,c_2,...,c_m$ of clauses on a finite set U of variables such that $|c_i| = 3$ for $1 \le i \le m$.
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') $\le$ 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 check in polynomial time that G' has evacuation time B or less.

We transform 3SAT to CONTRAFLOW. Let U = $\{u_1,u_2, ...,u_n\}$ and C = $\{c_1,c_2,...,c_m\}$ 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

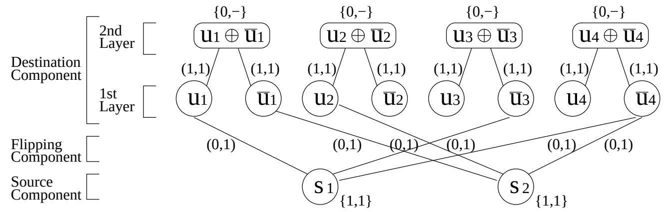

Figure 3: CONTRAFLOW instance resulting from 3SAT instance in which U = $\{u_1,u_2,u_3,u_4\}$ and C = $\{\{u_1,\overline{u}_3,\overline{u}_4\},\{\overline{u}_1,u_2,\overline{u}_4\}\}$.

source and destination components. The source component consists of vertices $s_1,s_2,...,s_m$ 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., $u_1,\overline{u}_1,u_2,\overline{u}_2,...,u_n,\overline{u}_n$). Second layer consists of XOR of each pair of literals (i.e., $u_1 \oplus \overline{u}_1, u_2 \oplus \overline{u}_2, ..., u_n \oplus \overline{u}_n$). This XOR layer serves as a destination node set in the CONTRAFLOW problem. The two nodes in a pair (u_i and $\overline{u}_i$) in the first layer are connected to each XOR node ($u_i \oplus \overline{u}_i$) 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 3 shows an example of the contraflow graph obtained when U = $\{u_1,u_2,u_3,u_4\}$ and C = $\{\{u_1,\overline{u}_3,\overline{u}_4\},\{\overline{u}_1,u_2,\overline{u}_4\}\}$.

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

$\rightarrow$: 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.

$\leftarrow$: Suppose that *EVAC-TIME*(G') $\le$ 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.

3. RELATED WORK

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[5, 4]. A survey by [22] of evacuation issues and contraflow revealed that planners have no recognized standards or guidelines for the design, operation, and location of contraflow segments. Many states threatened by hurricanes and considering contraflow plans were dependent on past evacuation experiences.

Past papers and Department of Transportation reports[22, 21, 20, 6] have mainly tackled the managerial and operational aspects of contraflow such as signal control, merging and cost. When planners design network configuration for evacuation scenarios, they mainly depend on past experiences and guesses. Such approaches may be effective to decide the direction of main highway contraflows. However, it is almost impossible to depend on past experience alone to deal with a large transportation network with changing demographic data and temporal factors such as construction zones.

[14] introduced two different contraflow algorithms from a computer science perspective. One is based on a multicast routing problem and the other is based on breadth-first graph traversal. These algorithms can handle only a single-coordinated incident due to conflicts of multiple optimal paths preferred by different source nodes in case of multiple-sources and multiple-destinations evacuation model. They did not clearly describe the use of different link capacities in the construction of optimal routes. On the other hand, our heuristics can handle the multiple-sources and multiple-destinations evacuation situation which has inherently conflicting edges by more than a path from different sources. Our heuristics also explicitly use capacity as an important factor to decide road directions. Our proposed heuristics can consistently reduce evacuation time by more than a third. In addition, each heuristic has its own set of properties.

4. PROPOSED HEURISTICS

In this section, we introduce two different approaches to generate contraflow network configurations. The first approach is tailored to the properties of contraflow and flow theory using a greedy algorithm, while the other approach exploits combinatorial optimization technique.

4.1 Flip High Flow Edge (FHFE): A Greedy Approach

4.1.1 Time Expanded Graph and Minimum Cost Flow Theory

To understand the details of FHFE algorithm, it is necessary to have related background knowledge. Thus, two important concepts are described in this subsection: 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[12]. Given a directed network $G(V, E)$, we can define the time expanded graph G_T in the following way.

Definition 1. Let $G(V, E)$ be a directed network with V the set of vertices with initial occupancy V_{occ} and vertex 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.
$V_T := \{V_{id,t} \mid id \in$ vertex id of V and $t = 0, 1, \cdots, T\}$
$E_T :=$
$\{(V_{id,t},\ V_{id,t+1}) \mid id \in$ vertex id of V and $t = 0, \cdots,$ T-1$\}$
$\cup\ \{(V_{id1,t1},\ V_{id2,t2}) \mid id1,\ id2 \in$ vertex id of $E_{id1,id2}$ and $t2 = t1 + \lambda_{id1,id2}\}$
$\cup\ \{(S,\ V_{id,0}) \mid id \in$ vertex id of V with $V_{occ} > 0\}$

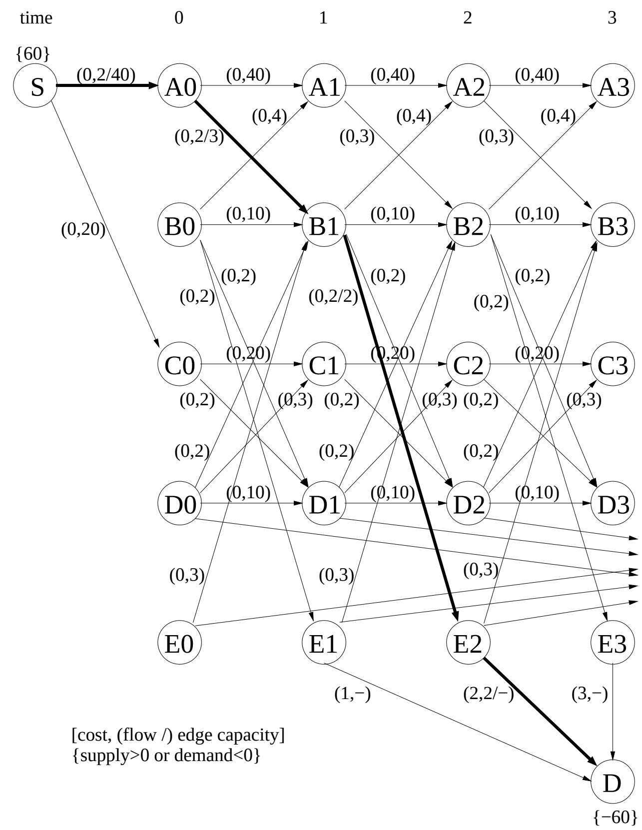

Figure 4: Time Expanded Version of the Graph in Figure 1

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

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 vertex V_{id} 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 vertices are moving from V_{id1} to V_{id2} in original network G. In Figure 4, 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 the 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 4 derived from Figure 1. The graph in Figure 1 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 4 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 original graph over time period from 0 to T (time horizon). At first, T should be set the value 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 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 vertices 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 2. Thus, only two evacuees can escape through the path. The travel times of edges (A,B) and (B,E) in 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 original graph. If we find a set of shortest paths based on available capacity, the set naturally becomes evacuation plan because each flow amount in the flow set corresponds to a group of evacuees escaping in minimal evacuation time. The description in this paragraph is based on minimum cost flow theory. The solution produced by minimum cost flow algorithm is always optimal in the sense that we can acquire the minimum evacuation time.

4.1.2 FHFE

The basic assumption of FHFE heuristic is that the edges more frequently used in the original network configuration for an evacuation plan help determine the selection of edge flippings. Therefore, we need to know the flow history of the original network configuration in advance by applying a minimum cost flow algorithm on the time expanded graph of the original network configuration. If a flow history value is assigned to each edge, we can choose the direction of two-way edges in favor of increasing outbound capacity in the following way. If one of the two-way edges in the original network configuration is used while the opposite edge is not used, we flip the opposite edge to increase capacity. If both edges are used in the original network configuration, we flip the edge which has less flow history.

Algorithm FHFE;
1. *Generate a time expanded graph from the given network;*
2. *Apply minimum cost static flow network algorithm on the time expanded graph;*
3. *Set flow history value from step2 to each edge and sort edges by flow values in descending order;*
4. *For the first m% of edges in the sorted edge set, change the direction of each edge toward the direction of larger flow;*
5. *Apply minimum cost static flow network algorithm to the configuration from step4 to get the final evacuation time;*

Edge	Flow	Edge	Flow
A-B	59	D-B	17
B-E	42	B-D	15
C-D	26	D-C	6
B-A	19	E-B	0
D-E	18	E-D	0

Table 1: Flow History of the Simple Network in Figure 1 used for FHFE

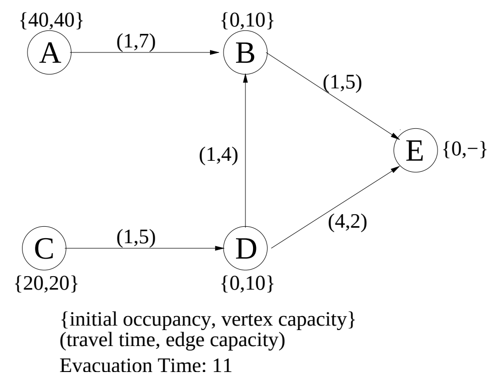

Figure 5: Contraflowed Configuration of the Simple Network in Figure 1 using FHFE

If we use an existing minimum flow program such as NETFLO[15] or RelaxIV[1], the program produces a flow history of each edge as well as a total cost value. In step 3, the flow history value is assigned to each edge in the original network configuration and edges are sorted by flow values in the edge set. Edges with more flow values tend to be more meaningful in the context of contraflow. Thus, we select only the first m% of such meaningful edges and flip them in favor of increasing outbound capacity. The selection percentage m is a degree of contraflow parameter. The effect of parameter m will be examined in the evaluation section.

The time expanded graph for our simple case in the problem definition section is illustrated in Figure 4. Table 1 shows the flow history generated by applying the minimum cost flow algorithm to the time expanded graph and sorted by flow values in descending order. If we send flow through (A,B) several time in original graph (e.g., flow 2 through A-B-E and flow 1 through A-B-D-E, etc.), the cumulative value will be 40 until the initial occupancy 40 empties. Someone may wonder why flow between A and B is 59, not 40 in Table 1. It is because some portion of flows oscillated between A and B. The flow 19 of edges (B-A) offsets flow 59 of edges (A-B). Suppose m is 70%. Then, we can flip the first seven edges A-B, B-E, C-D, B-A, D-E, D-B and B-D in support of increasing capacity. Especially for edges between B and D, the flow from D to B is greater than from B to D.

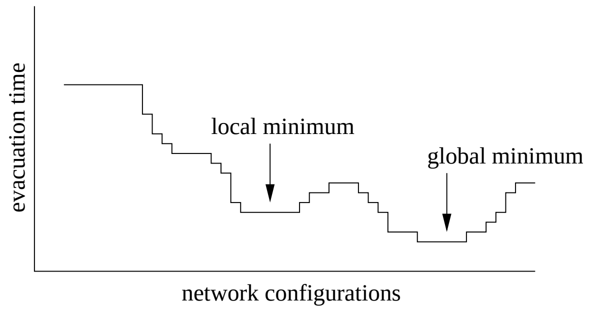

Figure 6: State Space Landscape of A Contraflow Problem

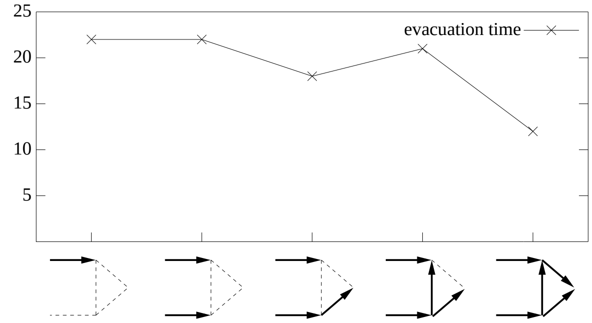

Figure 7: Possible Iterations of the SA for the Simple Network used in Figure 1

Thus, we fix the edge from D to B. The resulting configuration is shown in Figure 5. With the new configuration, the evacuation time is reduced from 22 to 11.

For the run time analysis, suppose that there are n vertices and m edges in the original network G with initial occupancy p. The run time of step 1 and 4 in the FHFE is trivial. The step 3 takes $O(m\log m)$ if we use well-known sorting algorithms like heap, merge or quick sort. Thus, the dominant run time part of the FHFE is step 2 and 5 where we apply minimum cost maximum flow algorithm to the time expanded network G_T. Let T be the time horizon, or maximum evacuation time. Then, the upper bound of nodes in G_T is $N = n(T + 1)$. The upper bound of edges in G_T is $M = (n + m)T + m$ - $\Sigma_{(i,j)\in m}\lambda_{ij}$ where λ_{ij} denotes travel time of edge (i,j)[12]. The asymptotically fastest method for finding a min-cost max-flow is designed by [19] and runs in $O(M\log N(M + N\log N))$ in G_T.

If we assume that the transportation network is sparse with the average degree of vertices 3, we can consider that $m = (3{*}n)/2 = 1.5n$. We can also assume that the maximum evacuation time T is proportional to the occupancy value p. Then, we can assume that N is proportional to np and M is also proportional to np. Thus, the above run time can be reduced in the following way.

$O(M\log N(M + N\log N))$
$= O(np\log np(np + np\log np))$
$= O(n^2p^2\log^2(np))$

4.2 Simulated Annealing (SA)

If we can interpret the contraflow problem as NP complete, we can focus on finding a "good" solution rather than trying to find an optimal solution. If we assign the evacuation time to a given system as a cost function or objective function, the "goodness" of a contraflowed network can be quantitatively measured. Thus, if we draw a state space landscape with evacuation times, our problem is equivalent to the search problem of finding the lowest valley (see Figure 6). In other words, we can interpret the contraflow problem as a combinatorial optimization problem due to its NP complete characteristic.

Simulated annealing (SA) is a general purpose combinatorial optimization technique formulated by Kirkpatrick et al. in 1983[18]. The SA method was created as an analogy to annealing process of liquid cooling with the goal of escaping from local minima on a combinatorial search landscape. Many applications have been reported to utilize SA as a tool to solve large-scale combinatorial optimization problems, from a typical traveling salesman problem to complicated chip layout design.

The new perturbation is accepted when the new objective function value improves or the gap between the old objective function and new objective function of inferior perturbation is within the formulated probability. By accepting some inferior perturbations, SA opens the path to a global optimum. The SA algorithm has the following structure[18].

procedure SimulatedAnnealing;
1. $S := S_0$; {initial solution}
2. $T := T_0$; {initial temperature}
3. *iterations* := i_0; {initial # of iterations for inner loop}
4. *repeat*
5. *repeat*
6. $NewS := perturb(S)$;
7. *if* $h(NewS) < h(S)$ *or random* $< e^{(h(S)-h(NewS))/T}$
8. *then accept* := *true*;
9. *else accept* := *false*;
10. *until inner loop has been repeated iterations times*;
11. $T := \alpha{*}T$; *iterations* := β**iterations*;
12. *until out of time*;

(h: objective function, S: current state,
$NewS$: new perturbation,
random: pseudo-random number in the range [0,1],
T: cooling temperature)

To apply SA to our contraflow problem, we configure the following factors. First, the initial state is the original network configuration for evacuation planning. Second, our perturbation is based on flipping an edge (↑↑, ↓↑ and ↓↓). Third, the objective function is evacuation time. Fourth, the cooling schedule and termination condition can be configured with experiment parameters. Finally, the order of flipping is another random factor during simulation.

It is known that SA is not always able to find the global optimum, but can find minimum values that are close to the global minimum. In addition, the search landscape should not have an overly steep curve. That is, the width of valleys should be approximately proportional to their depth. We will consider these elements in the evaluation section and examine how the contraflow search problem is suitable to the SA method.

Figure 7 is an illustrative example to show how the SA can behave as each edge is flipped in the graph used in Figure 1. First two flips do not change evacuation time and draw

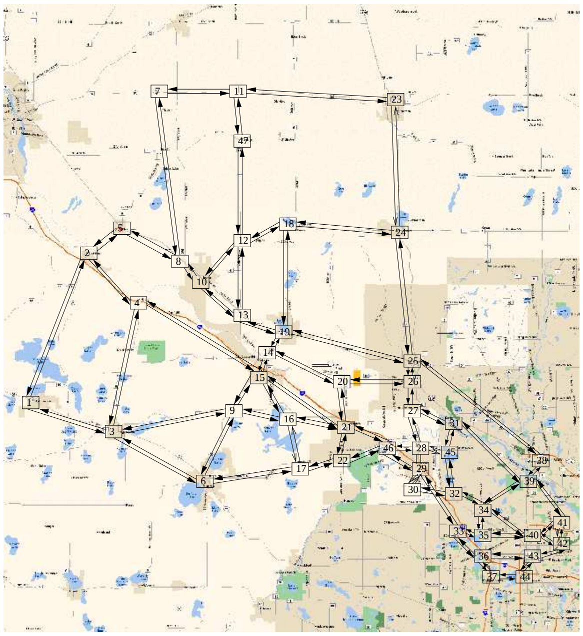

Figure 8: Nuclear Power Plant Case Map

a plateau. A change from the third to the fourth configuration causes inferior movement which is allowed in the SA algorithm. In the fifth configuration, the evacuation time finally reaches the optimal state.

5. EVALUATION

5.1 Experiment Setup

We implemented and evaluated the algorithms presented in this paper. The language used was C++ and the experiments were performed on a dual CPU Pentium III 650MHz workstation with 2GB of memory, running Linux. A program representing graph structure was implemented with Java.

Figure 8 shows a virtual scenario of a nuclear power plant failure in Monticello, Minnesota. There are fourteen cities around the facility and one destination shelter. The demographic data are based on the Census 2000 population data. The total number of evacuees is about 42,000. If the given situation is converted to a graph representation, the graph has 47 vertices with 148 edges. This dataset seems to be trivial size compared with large transportation network we have seen. However, we need to select adequate size of dataset to handle iterative experiments of simulated annealing. The interstate highway (path 2,4,15,21,45,29,33,36 and 43) has larger capacity than other edges. The destination is vertex 40 on the bottom right corner of the map. The evacuation time with the original network configuration is 272 min (4 hr 32 min).

We also implemented a network generator in Java to conduct scalability experiments. The input file format to generate a network is similar to that of NETGEN[16] which generates transportation network with capacity constraints and initial supplies based on input parameters. When we changed a network size, the proportion between the number of nodes and edges was maintained according to the

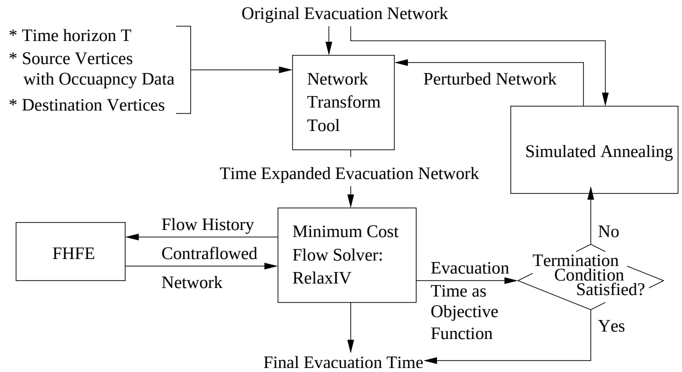

Figure 9: Experiment Design

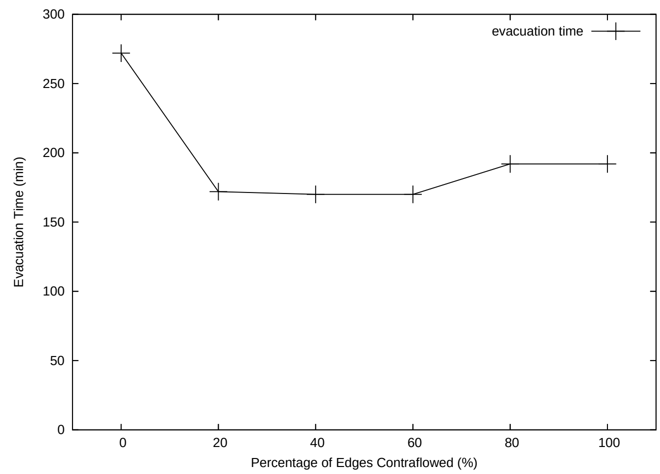

Figure 10: Evacuation Time with respect to Degree of Contraflow using FHFE

Monticello network. Edge lengths and capacity values were randomly generated within the range of Monticello network. Therefore, the resulting synthetic networks were close to real networks.

Figure 9 describes the experiment design to apply our algorithms to the Monticello nuclear power plant case. First, the original evacuation network is given with source and destination vertices. The network transform tool converts the input network into a time expanded graph with a given time horizon T[12]. Second, we use RelaxIV as a minimum cost flow solver. We can acquire flow history and evacuation time from the solver. The FHFE algorithm takes the flow history as input and generates a contraflowed network. In the simulated annealing algorithm, the original network configuration is used as an initial solution. The perturbed network means a newly generated network configuration by flipping an edge at each iteration step. The evacuation time from RelaxIV is used as the objective function value to assess the perturbed network. When the iterative process satisfies a certain termination condition, it generates a final evacuation time.

5.2 Flip High Flow Edge (FHFE)

Figure 10 shows the results of applying the FHFE algorithm. The x axis represents the degree of contraflow and the

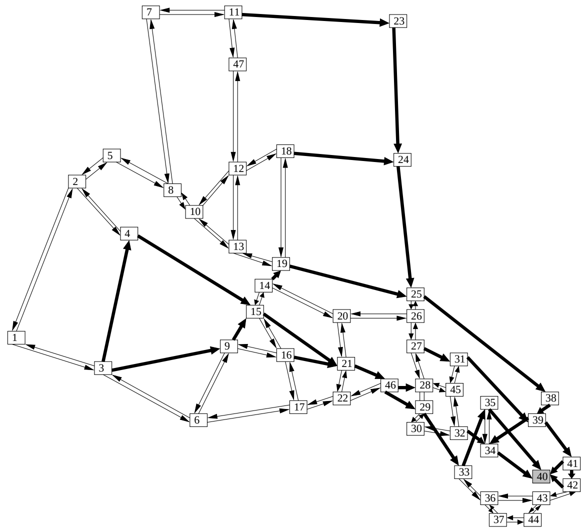

Figure 11: Configuration of 20% Contraflow Degree using FHFE

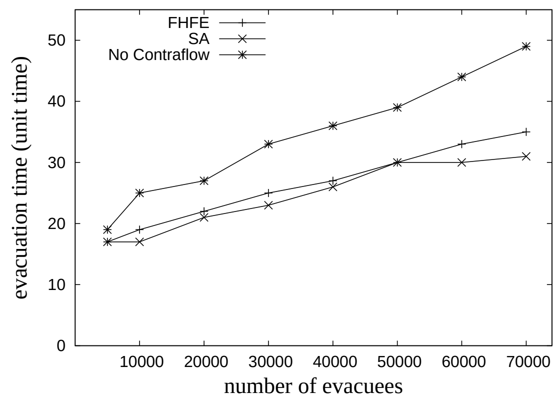

Figure 12: Evacuation Time With Respect To Number of Evacuees

y axis indicates evacuation time. The degree of contraflow parameter m described in the algorithm description section selects the first $m\%$ of total edges. The best result is 170 minutes at 40% and 60% degree of contraflow. When the percentage of edges contraflowed is 0%, it is equivalent to the evacuation of the given original network configuration. We observe that there is a steep improvement at 20% of contraflowed edges. The FHFE algorithm chooses the most promising edges and then contraflows them first. Therefore, it is natural to observe such an improvement with a small percentage of contraflow degree. Another interesting observation is that a high percentage of contraflow degree does not always lead to better evacuation time. This means that some edges should be used with both directions rather than contraflowing them.

Figure 11 is a snapshot of network configuration when the degree of contraflow is 20%. We can observe that edges with high capacity and edges close to the destination are contraflowed.

5.3 Simulated Annealing (SA)

Experiments were performed with three different initial temperatures 20, 10 and 5. In all cases, the temperature update constant α was set to 0.9. The iteration increasing factor β was set to 1.1 and the inner loop started from 50 and the outer loop terminated at 10. For each initial temperature, 30 different random seeds were applied to change the order of flippings.

Two random seeds at initial temperature 5 generated evacuation time 166 minutes, which is better than that of the FHFE algorithm. These results can be interpreted as being close to a global minimum by escaping from several local minima with the help of the simulated annealing characteristic. Interestingly, the lowest initial temperature produced the best result, which is not the case generally in simulated annealing. The search space landscape of contraflow configurations consisted of wide plateaus and sudden jumps of the objective function. Therefore, a significant level of stochastic factors prevented the experiment from configuring appropriate parameters.

5.4 Scalability

In the scalability experiment, we generated synthetic transportation networks with sizes proportional to the Monticello case. When we increased a network size, the proportion between the vertices and edges corresponded to that of the Monticello case. In addition, randomly generated edge length and capacity values were also within the range of the Monticello case. Thus, we were able to generate networks close to a real situation. In this experiment, we wanted to tackle two scalability issues.

- **Effects of Number of Evacuees** The purpose of this experiment was to evaluate how the number of evacuees affects evacuation time. We fixed the total number of vertices as well as the number of source and sink vertices. Only the number of evacuees was varied from 5,000 to 70,000. Figure 12 presents the results regarding evacuation time. The evacuation time linearly increases for both the original network and contraflowed network. Although the SA approach produced better evacuation times, the FHFE also produced evacuation times close to those of the SA.

- **Running Time with respect to Network Size** In this experiment, we evaluated how the network size affects the performance of the algorithm. The elements including vertices, edges and the number of evacuees changed proportionally with the number of vertices. Figure 13 presents the results. The iterative property of SA prevented us from trying big networks. Only networks with vertices from 50 to 150 were tested. Our FHFE algorithm showed almost zero run time with small networks while the SA showed rapidly increasing run time. However, FHFE also showed quadratic run time with respect to number of vertices when we increased the size of networks to 3000. This result corresponds to the run time analysis of the FHFE.

6. DISCUSSION

In this section, we discuss related knowledge to understand our analysis of the contraflow problem. Specifically,

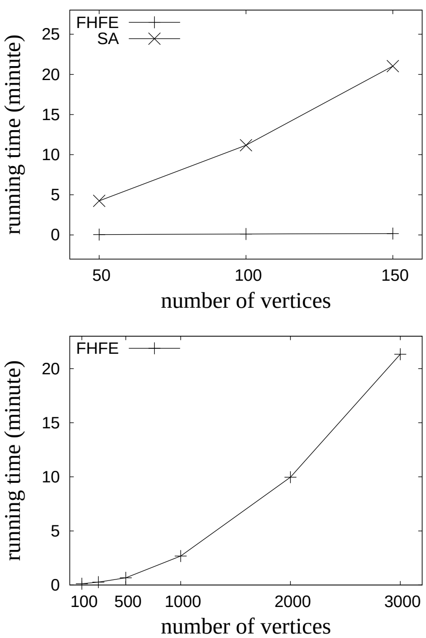

Figure 13: Running Time With Respect To Network Size

our work is related to two research problems. One is mathematical analysis of evacuation planning. The other is analysis of combinatorial optimization. Previous works on mathematical analysis of evacuation planning have focused on macroscopic approaches using flow models[7, 13]. [12] studied a discrete time dynamic network flow model with algorithms for evacuation planning in general. [17] proposed a close-to-optimal solution for evacuation planning while significantly reducing computational cost. Their fast algorithm can be combined with ours in the future work. NETFLO[15] and Relax4[1] are famous solvers for linear minimum-cost flow problems. Especially, Relax4 contributed to the construction of our algorithms by quickly calculating the evacuation time of a given network using dual ascent methods and auction algorithm.

To utilize general combinatorial optimization heuristics to solve a specific problem requires much experience. Thus, it is necessary to read various applications in the domain. [18] formulated a class of adaptive heuristics for combinatorial optimization including probabilistic hill-climbing and simulated annealing. [8] provides a broad overview of NP complete problems and their proofs. [3] studied the application of simulated annealing to the drawing of graphs nicely. The paper gave valuable information from their trial and error. [10, 11] presented a broad overview in the domain of annealing. Our work applies the insights from these previous studies to solve contraflow problem using general combinatorial techniques.

With the discussion of related knowledge, it is worth exploring possible solutions of contraflow problem. General search algorithms such as A* have difficulty in defining the goal state of contraflow problem because the optimal network configurations leading to minimal evacuation time exist in a set of combinatorial possibilities with unknown minimal evacuation time. Dynamic programming algorithms such as branch and bound approach cannot be applied because contraflow problem does not guarantee sub-optimality between the given stage and its sub-problems. General search techniques such as hill climbing can improve the given state, but mostly lead to the local minima.

Algorithm	Evacuation Time
Original Configuration	272 min
FHFE	170 min
Simulated Annealing	166 min

Table 2: Evacuation Time Comparison on Monticello Nuclear Power Plant Case

7. CONCLUSION AND FUTURE WORK

Current evacuation procedures heavily depend on the use of surface traffic through the limited capacity of road networks. From this perspective, contraflow becomes one of the key solutions of evacuations on the existing transportation infrastructure. Due to the nature of transportation networks, we modeled evacuation situation using graph data structure and analyze it with flow theory. In our model, one or more source nodes can be added, whereas existing algorithms only cover the single source situation due to conflicts of optimal paths from different source nodes. Multiple-sources and multiple-destinations contraflow problem belongs to a category of NP completeness problems. Our main contribution lies in the fact that our proposed approaches efficiently address the NP complete contraflow problem with the cost model and experimental validations. In this paper we presented two contraflow approaches and their evaluations with both real and synthetic networks. Here is a brief summary of the approaches.

- **FHFE** generates promising results in spite of its fast run time. Evacuation planning software needs to be interactive due to various combinations of input parameters (e.g., daytime v.s. night time population) and changing datasets (e.g., construction zone). Thus, running time is a critical factor when we implement planning software for clients. A well designed heuristic algorithm that is tailored to contraflow problems has some advantages over general iterative methods. The number of contraflowed edges is adjustable. The scalability of FHFE is superior to that of the SA approach.

- **Simulated Annealing** reveals the possibility of the existence of a global minimum. In addition, the application of SA to the contraflow problem has never been observed in optimization research area. Selecting the appropriate perturbation and parameter values for experiments is of critical importance in simulated annealing.

Table 2 summarizes the evacuation time of our approaches for the Monticello case. We can observe that both approaches significantly reduce evacuation time and that the FHFE can generate a contraflow network whose evacuation time is competitive with that of the SA.

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 would 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. For example, we can flip 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. Finally, a more extensive parameter set of simulated annealing needs to be tested for in-depth knowledge of its applicability to contraflow problems.

8. ACKNOWLEDGMENTS

We are particularly grateful to members of the Spatial Database Research Group at the University of Minnesota for their helpful comments and valuable discussions. Especially, our evacuation research members including Qingsong Lu and Betsy George gave us valuable comments. Kimberly Koffolt increased the readability of our paper by proofreading it. We are also thankful for use of the program RelaxIV developed by the CRIFOR research group.

This work is supported by a grant (contract number 81655) from the Minnesota Department of Transportation.

9. REFERENCES

[1] D. Bertsekas and P. Tseng. Relax-iv: A faster version of the relax code for solving minimum cost flow problems. In *LIDS(Laboratory for Information and Decision Systems)-P-2276*. M.I.T., Cambridge, MA, 1994.

[2] B. V. Cherkassky, A. V. Goldberg, and T. Radzik. Shortest paths algorithms: theory and experimental evaluation. In *SODA(Symposium on Discrete Algorithms) '94: Proceedings of the fifth annual ACM-SIAM(Society for Industrial and Applied Mathematics) symposium on Discrete algorithms*, pages 516–525. Society for Industrial and Applied Mathematics, 1994.

[3] R. Davidson and D. Harel. Drawing graphs nicely using simulated annealing. *ACM Transactions on Graphics*, 15(4):301–331, October 1996.

[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 Network*. Princeton University Press, 1962.

[8] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP Completeness*. Freeman, 1979.

[9] R. Greiner. Probabilistic hill-climbing: Theory and applications. *Proceedings of the Ninth Canadian Conference on Artificial Intelligence*, pages 60–67, 1992.

[10] B. Hajek. A tutorial survey of theory and applications of simulated annealing. *24th IEEE Conference on Decision and Control*, pages 755–760, 1985.

[11] B. Hajek. Cooling schedules for optimal annealing. *Mathematics of Operations Research*, 13(2):311–329, 1988.

[12] H. W. Hamacher and S. A. Tjandra. Mathematical modeling of evacuation problems: State of the art. *Pedestrian and Evacuation Dynamics*, pages 227–266, 2001.

[13] H. W. Hamacher and S. Tufekci. On the use of lexicographic min cost flows in evacuation modeling. *Naval Research Logistics*, 34:487–503, 1987.

[14] G. Hamza-Lup, K. A. Hua, M. Le, and R. Peng. Enhancing intelligent transportation systems to improve and support homeland security. In *In Proc. of IEEE International Conference on Intelligent Transportation Systems*, 2004.

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

[16] D. Klingman. A program for generating large scale capacited assignment, transportation, and minimum cost flow network problems. In *Management Science*, pages 20:814–822, 1974.

[17] Q. Lu, Y. Huang, and S. Shekhar. Evacuation planning: A capacity constrained routing approach. *ISI(International Symposium on Isotopomers) 2003*, pages 111–125, 2003.

[18] S. Nahar, S. Sahni, and E. Shragowitz. Simulated annealing and combinatorial optimization. In *23rd IEEE Design Automation Conference*, pages 293–299, 1986.

[19] J. Orlin. A faster strongly polynomial minimum cost flow algorithm. In *STOC '88: Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 377–387. ACM Press, 1988.

[20] G. Theodoulou. Contraflow evacuation on the westbound i-10 out of the city of New Orleans. Master's thesis, Louisiana State University, 2003.

[21] B. Wolshon. One-way-out: Contraflow freeway operation for hurricane evacuation. *Natural Hazards Review*, 2(3):105–112, 2001.

[22] B. Wolshon, E. Urbina, and M. Levitan. National review of hurricane evacuation plans and policies. Technical report, LSU Hurricane Center, Louisiana State University, Baton Rouge, Louisiana, 2002.