Maximum Throughput Flow-Based Contraflow Evacuation Routing Algorithm

Manki Min*, Jonguk Lee†
Dept. of Electrical Engineering and Computer Science
South Dakota State University
Brookings, SD 57007
Emails: manki.min@sdstate.edu*, jonguk.lee@jacks.sdstate.edu†

***Abstract*—With the combination of vehicular networks and a scalable algorithm, we can expect an effective real-time contraflow evacuation routing. In this paper we define the maximum throughput flow and propose a contraflow evacuation routing algorithm based on reverse shortest paths and maximum throughput flows that can be used for the real-time contraflow evacuation routing. The proposed algorithm computes the contraflow scheme by finding minimal number of shortest paths and hence its computation is highly efficient and scalable. In addition, the evacuation time for the computed contraflow scheme is better than or same as that of CCRP++ with the help of maximum (or at least higher) throughput flows. The computational results confirm the efficiency of the computation and the effectiveness of the computed contraflow schemes.**

I. Introduction

As a means of an efficient evacuation by maximal utilization of road lanes, contraflow evacuation routing algorithms are getting more and more attentions. As opposed to the case where static contraflow scheme is predetermined, for the real-time evacuation routing we need a dynamic contraflow scheme. And considering the urgent need of the evacuation routes, the computation of a proper contraflow scheme should be done as quickly as possible. With the help of vehicular networks and wireless networks, we can envision that this contraflow scheme is directly delivered to the drivers of cars on the road. In this situation, we can expect highly more efficient evacuation process by skipping the physical implementation of the computed contraflow schemes on the roads.

The contraflow routing problem is, given a transportation network graph where edges have travel time and capacity and nodes have initial occupancy, to find a contraflow scheme, i.e. reversing some of the edges, that minimizes the evacuation time. Hence path-finding is an essentially common underlying algorithm for any contraflow computations. In some cases, it can be a shortest path; in some others, a quickest path. Hence in order to reduce the computation time, the most important factor is the path-finding time. In this work, we present a new contraflow evacuation routing algorithm that is based on reverse shortest paths. This way we can run only one shortest path-equivalent algorithm and find all the shortest paths from all source nodes to destination nodes and therefore reduce the path-finding time significantly.

Our contribution in this work is two-folded: i) we define the maximum throughput flow that results in a minimized evacuation time and ii) we present a new contraflow evacuation routing algorithm based on the maximum throughput flows. When we consider the whole evacuation process as a continuous flow, the throughput determines the evacuation time and maximum flow is not always the maximum throughput flow. Rather, it needs additional information such as the times when the first and last evacuees arrive at a destination using a path and a maximum flow does not guarantee the maximum throughput. From the concept of the maximum throughput flows, we can construct a new efficient algorithm for contraflow evacuation routing.

The rest of this paper is as follows: Section II will define the problem of contraflow evacuation routing and discuss briefly the existing algorithms. In section III, the novel concept of maximum throughput flow is explained and discussed. The algorithm MTFC (Maximum Throughput Flow-based Contraflow) is introduced in section IV and explained. The computational results are presented and discussed in section V and section VI concludes this paper.

II. Related work

The material and literature on evacuations and the contraflow problem have been published in various domains, including not only transportation, but also social and behavioral sciences, and mathematics [2], [3]. A survey [24] 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 get more and more threatened by hurricanes and other disasters and they consider contraflow plans that are dependent on past evacuation experiences. Planning problems of hurricanes Katrina and Rita were identified by Litman [12] and they specifically criticized the unplanned contraflow orders and failure to use contraflow lanes. Past papers and Department of Transportation reports [7], [19], [23], [24] have mostly discussed the operational and managerial aspects of contraflow such as merging, signal control and cost. When planners design network configuration for evacuation scenarios, they tend to depend on empirical guesses or previous evacuation records. Such handcrafted contraflow plans have revealed that they are neither flexible to accommodate various variables nor efficient to find critical road segments of contraflow [5]. Hamza-Lup et al. [9] introduced two contraflow algo-

978-1-6128-4937-9/13/$31.00 ©2013 IEEE

rithms from a perspective of computer science; one based on a multicast routing algorithm and the other based on the breadth-first graph traversal. These algorithms can handle only single source node cases due to the conflicts between multiple optimal paths that occur in multiple-source and multiple-destination evacuation models. In addition, the use of different link capacities was not clearly described. Thus, their approach is not effective under certain circumstances for example when the number of travelling units is finite, or road capacities are constrained, or specific destination nodes are prescribed, or evacuees are spread over many locations. Tuydes and Ziliaskopoulos [21] proposed a mesoscopic contraflow network model based on a dynamic traffic assignment method. They formulated the capacity reversibility using a mathematical programming method. However the discretized hypothetical network required solving the traffic assignment problem prohibits large scale network scenarios from running in their framework. They also tried a Tabu-based heuristic approach [22] to address the proposed capacity reversibility optimization. Their solutions only worked with small-size network input since the computation required a considerable number of iterations. Theodoulou and Wolshon [20] used CORSIM microscopic traffic simulation and modeled the free way contraflow evacuation around New Orleans. They were able to suggest alternative contraflow configurations at the detailed levels with the help of the micro scale traffic simulator. However, the microscopic simulation model requires a manpower-intensive network coding and huge running time for each scenario, making it difficult to take advantage of spatial databases or easily compare alternative configurations. Evacuation route planning with other microscopic traffic simulation has shown similar limitations.

In many cases, the evacuation routing algorithm is used as a part of an algorithm to solve another problem such as the contraflow evacuation planning problem [10], [11]. The Greedy algorithm in [11] needs complete evacuation routing plan to get the contraflow configuration to minimize the evacuation time. The BottleneckRelief algorithm in [11] does not need the evacuation routing plan however the Bottleneck-Relief algorithm is not more time-efficient than the Greedy algorithm since it depends on the expensive maximum flow calculation. Moreover the evacuation time is shorter with the Greedy algorithm than the BottleneckRelief algorithm, so we use the Greedy algorithm for our comparison. Regarding the evacuation routing, most algorithms are based on CCRP [13]. CCRP++ [25] is an improved version of CCRP by reusing the shortest paths that are already found and gets significant improvement in the computation time.

The concept of throughput in dynamic network flows with infinite time horizons was introduced by Orlin [18]. In this case the flows are considered to be circulating in the network which is different from our situation. In the infinitely circulating network, the throughput of flow in a path is simply the capacity of the path and we don't need any time-lined information. However, when we consider a finite non-circulating network, the throughput of flow in a path becomes smaller than the capacity of the path. Our algorithm is based on more accurate definition of throughput of flows and it provides better results than other algorithms such as Greedy algorithm [11] using significantly reduced computing time. In the next section, we define our version of throughput of flows and discuss about the maximum throughput flows upon the definition.

III. Maximum Throughput Flows

In this section, we define maximum throughput flow which is the basis of the proposed algorithm.

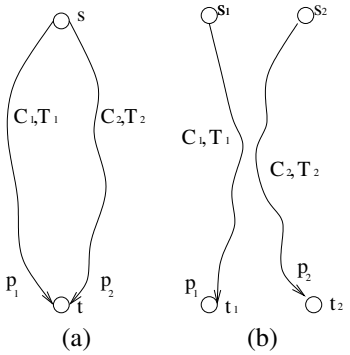


Fig. 1. Single source and multiple sources cases

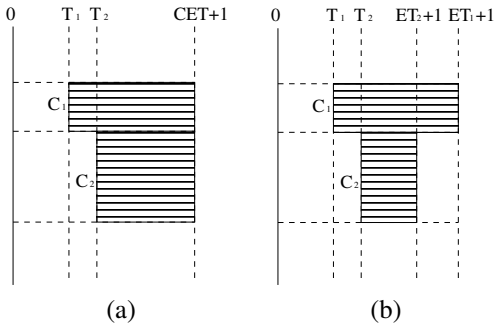


Fig. 2. Arrival graphs for single source and multiple sources

The throughput of the flow on a path can be defined as the ratio of the total amount of the flow that goes through the path over the time interval. If the flow is circulated inside the network infinitely, then the throughput is simply the total amount of the flow. However the network that we deal with in this work is neither a circular network nor an infinite flow. In this case, the time period should include every time period when the flow goes through an edge in the path. In other words, the throughput of flows on an evacuation path is the total amount of flow over the evacuation time. When we track the incoming flow to a destination node, we will see some portion with no flow on the incoming edges, but during that period, the flow goes through other edges in the path and hence that period should also be included. We call the graphs that keep track of incoming flow to destination nodes based on time as in Figure 2 as arrival graphs.

Consider the following two cases: a single source node case and a multiple source nodes case as in Figure 1. As in (a),

considering the two paths P_1 and P_2 from the single source node S, we get the arrival graph at the destination node as in Figure 2 (a). In this simple case, the evacuation times from each path will be the same as the CET (Combined Evacuation Time) using the two paths and hence the throughput of the flow is

$$\frac{C_1*(CET+1-T_1)+C_2*(CET+1-T_2)}{CET+1}$$

$$= C_1 + C_2 - \frac{C_1*T_1+C_2*T_2}{CET+1}.$$

The third term of the last line of the previous formula accounts for the wasted area in the arrival graph and in order to maximize the throughput we need to minimize this wasted area and maximize the sum of capacities. Therefore allocating as full capacity as available onto the shorter paths will result in a maximum throughput flow scheme and this approach can even be easily implemented.

Now looking at the second case of multiple source nodes as in Figure 1 (b), we get the arrival graph as in Figure 2 (b). In this case since both evacuation times E_1 and E_2 are not equal, the throughput of the flow is

$$\frac{C_1*(ET_1+1-T_1)+C_2*(ET_2+1-T_2)}{ET_1+1}$$

$$= C_1 + \frac{C_2*ET_2+1}{ET_1+1} - \frac{C_1*T_1+C_2*T_2}{ET_1+1}$$

$$= C_1 + C_2 - \frac{C_1*T_1+C_2*T_2}{ET_1+1} - \frac{C_2*(ET_1-ET_2)}{ET_1+1}.$$

The third term above, which is essentially same as that in the single source node case, still accounts for the wasted area in the arrival graph that appears before the first arrival and the fourth term accounts for the wasted area after the last arrival from the source node s_2. If the arrival graph has some discontinued flow, then the throughput becomes even smaller than the previous formula. Again even in this case, allocating as full capacity as available onto the shorter paths helps reducing the third term and in most cases it also helps to increase the throughput. Of course in this case, there is no guarantee that this capacity allocation scheme results in a maximum throughput flow. In order to calculate more accurate throughput of the flow, we need more information such as the time-lined utilization information of each edge, for example arrival graph-equivalent graphs for each edge, which requires expensive computations.

Assuming that we obtain a maximum throughput flow scheme for the evacuation, we can expect the minimum evacuation time. In multiple source node cases, allocating as full capacity as available to shorter paths still generates a flow scheme with higher throughput than other flow schemes and hence we can expect smaller evacuation time than other flow schemes. We use this anticipation in our algorithm design which is explained in the next section.

IV. Maximum Throughput Flow-based Contraflow (MTFC)

Our novel algorithm MTFC (Maximum Throughput Flow-based Contraflow) is presented in Figure 3.

Before we begin with the algorithm, we introduce the definitions used in the algorithm.

- A *reverse shortest path* from a node d to a node s is a shortest path from s to d and it can be computed by using the edges of reverse directions in a general shortest path algorithm such as Dijkstra's.
- e represents an edge.
- $e.S$ represents the source node of e.
- $e.D$ represents the destination node of e.
- $e.R$ represents a reverse edge of e such that $e.S = e.R.D$ and $e.D = e.R.S$.
- $e.C$ represents the capacity of e given in the problem instance.
- $e.AC$ represents the available capacity of e and is computed by the sum of the capacities of the edge e and $e.R$. If $e.R$ does not exists, $e.AC = e.C$.
- $e.DC$ represents the capacity determined by a reverse shortest path of e.
- $e.FC$ represents the final capacity determined and allocated by the algorithm on e.

Note that by finding a reverse shortest path from a super destination node (a virtual node not on the given graph that has zero travel time, infinity capacity edges from each destination node), we can get all the shortest paths from every node to a destination node at once just as getting all the shortest paths from the source node to every node at once by finding a single-source shortest path from the source node. Hence one reverse shortest path finding provides us all the shortest paths from every source node to destination nodes. This approach not only reduces the computation time, but also elegantly handles the multiple shortest paths that share an edge. Shared edges in multiple paths were one of the intricate difficulties that we faced when dealing with the multiple source node cases. Fundamentally speaking, the idea of starting from the destination nodes instead of the source nodes was the key of eliminating this difficulty.

The output of the algorithm MTFC is the final capacities that are allocated to the edges ($e.FC$ for each e). Based on the final capacities, we can easily determine the amount of contraflow on each edge. In the Capacity Allocation phase, we used a shortest path algorithm based on Dijkstra's on the edges of reverse directions. In other words, instead of checking outgoing edges from the current node for the relaxing, we checked incoming edges to the current node. It is clear that the reverse shortest path algorithm has equivalent computational complexity as the shortest path algorithm. Hence the computational complexity of the algorithm MTFC is the multiplication of the computational complexity of the shortest path algorithm and the number of possible shortest paths that we can find from the graph. We argue that our algorithm finds the shortest paths minimal times and hence the computational complexity of our algorithm is minimal among any algorithms that consider every possible path.

The optional phase of Available Capacity Allocation allocates the remaining available capacities back to the edges. The

Initialization

1: for each edge e:
2: reset $e.FC$ and $e.DC$, i.e., $e.FC = 0$, $e.DC = 0$.
3: if $e.R$ exists,
4: $e.AC = e.C + e.R.C$.
5: else
6: $e.AC = e.C$.

Capacity Allocation

1: while a reverse shortest path from the super destination node to the super source node is found using available capacities of edges:
2: for each source node s:
3: Determine the minimum available capacity C of edges on the path from s to a destination.
4: for each edge e on the path:
5: $e.DC = \max(e.DC, C)$.
6: for each edge e with non-zero $e.DC$:
7: $e.FC$ += $e.DC$, $e.AC$ -= $e.DC$.
8: if $e.R$ exists,
9: $e.R.AC$ -= $e.DC$.
10: for each edge e:
11: reset $e.DC$.

Available Capacity Allocation (optional)

Allocation of half available capacities (HA):

HA-1: for each edge e with $e.FC = 0$:
HA-2: if $e.R$ exists,
HA-2: $e.FC$ += $\lfloor e.AC/2 \rfloor$, $e.R.FC$ += $\lceil e.AC/2 \rceil$, reset $e.AC$ and $e.R.AC$.
HA-4: else
HA-5: $e.FC$ += $e.AC$, reset $e.AC$.

Allocation of full available capacities (FA):

FA-1: for each edge e with $e.FC = 0$:
FA-2: $e.FC$ += $e.AC$, reset $e.AC$.

Fig. 3. Algorithm MTFC

optional phase (HA) allocates half of the available capacities to all the two-way edges allocates full available capacities to all the one-way edges and hence the number of edges with zero capacity allocated will be minimized. The optional phase (HA) will allocate more capacities to the shortest paths, but edges not on the shortest paths will also get capacity allocated. Another optional phase (FA) allocates full capacity to the edges with no capacity allocated at the end of Capacity Allocation phase. We can expect a certain level of perturbation to deviate the local minimum by leaving the most opportunities for non-shortest paths to be found. Without this phase, the final capacity may be zero for some edges which means the removal of the corresponding edge. With the removal of some edges, we can expect more condensed and focused evacuation routing plan which will be helpful in real situation in a sense that it will reduce the confusion of the evacuation routes in the perspective of the evacuees.

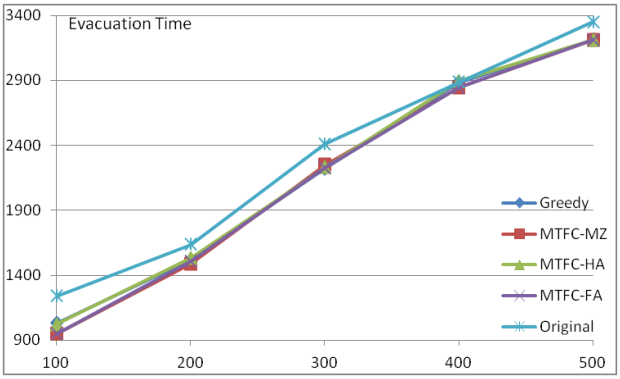
Our algorithm repeatedly finds the reverse shortest path and allocates the maximum available capacity on the edges included in the shortest paths. This allocation is made following our observation described in the previous section. Even if this maximum capacity allocation cannot guarantee the maximum throughput flow, it is highly probable that the allocation provides high throughput flow that leads to reduced evacuation time. The difficulty here is that we do not have any preprocessed information such as how much flow will go through an edge. In some cases, an edge with bigger capacity may have flow for only short period and the capacity is wasted for most of the evacuation time interval. If this happens, the capacity allocation does not result in the maximum throughput flow. When we put more emphasis on the evacuation time than on the execution time, such as in the case when the contraflow schemes are pre-calculated, we can think of combining MTFC with other evacuation routing planner and use the dynamic flow information to better allocate capacities to edges with more dynamic flows.

V. Computational Results

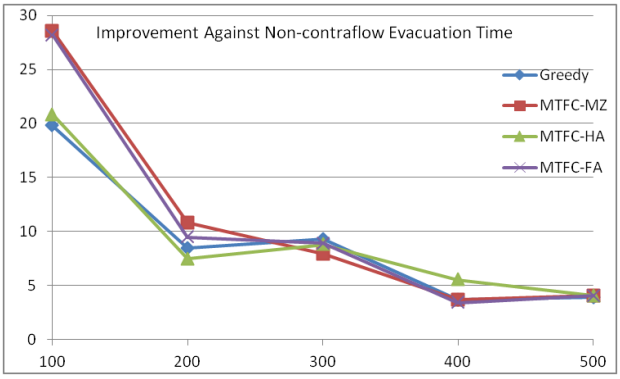
We implemented and compared two algorithms, Greedy algorithm in [11] and MTFC. For the evacuation routing, we implemented CCRP++ in [25]. For comparisons, we randomly generated n nodes in an $n \times n$ area with random occupancy for each node, with the value of n in $\{100, 200, 300, 400, 500\}$. The number of the source nodes, m_s, and the destination nodes, m_t, are randomly determined between 1 and 10 and between 1 and 5, respectively. Then the point of the disaster is randomly generated and the m_s nodes that are closest to the disaster point are marked as source nodes and the m_t nodes that are farthest from the disaster point are marked as destination nodes. For each source node, we randomly picked edges to generate at least one path. Up to $1.5 \sim 3$ times n edges are generated randomly. Each edge is assigned random capacity in between 1 and 5 and travel time proportional to the distance between two endpoint nodes. About 95% of the generated edges are made bidirectional with the same capacity and travel time on both direction. And we generated and ran 15 different instances for each transportation network setting. For the computation we used gcc and g++ as compilers on a Linux machine with dual 2.33 GHz dual core CPU's and 4GB of RAM.

The label Greedy represents the results of the Greedy algorithm in [11], MTFC-XX represents the results of MTFC, where XX being one of MZ (most zeroes, without optional phase), HA (half available capacity assigned), and FA (full available capacity assigned). As mentioned in the previous section, MTFC-MZ has the most zero-capacity edges and surprisingly the contraflow scheme generated this way provides pretty different results from the other contraflow schemes from time to time.

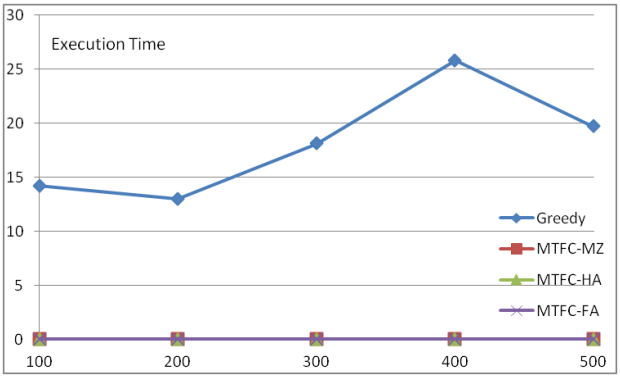
Figure 4 shows the averaged results of three contraflow schemes. (a) shows the averaged evacuation time based on the number of nodes and (b) shows the improvement of each contraflow scheme against the original non-contraflow scheme. As can be estimated from Figure 5, the Greedy algorithm tends to be the best in reducing the evacuation time. However considering the poor scalability, especially for bigger size of evacuees or smaller size of edge capacities, and the significantly higher computational complexity of the Greedy algorithm this result does not thrill us. Instead the results



(a)



(b)



(c)

Fig. 4. Comparison of average results

of any version of MTFC were very impressive, especially considering the computation time represented in (c). Moreover the computation time of any version of MTFC is highly scalable and depends only on the network size, not on other facts such as the number of evacuees or edge capacities.

Figure 5 shows the improvement in percentage of the evacuation times by contraflow schemes against the original evacuation time without any contraflows. In most cases, the four contraflow schemes provide very similar results in terms of the evacuation time. However, sometimes MTFC-MZ has extremely shorter evacuation time than others (especially Greedy), for example as in 14-the run of 100 node cases, 2-nd/13-th of 200 nodes, and 12-th of 400 nodes. In the mean time, MTFC-MZ also generates worse results as in 1-st of 200 nodes, 7-th/12-th of 300 nodes, 4-th of 400 nodes. This is an interesting result and needs more careful investigation, especially on the better result cases. Considering that CCRP or CCRP++ is a suboptimal algorithm to compute the evacuation time and recalling that MTFC-MZ will include only the edges that are included in the shortest paths, we can carefully argue that these worse results may simply show the ineffectiveness of CCRP-based algorithms. Or it could be the drawback of MTFC to heavily depend on the shortest paths, but in this case even with alleviation of this dependency in MTFC-FA still shows worse results as in 4-th of 400 nodes. Either case, more careful investigation about the actual evacuation plan and comparison with other results by different evacuation routing algorithms will be helpful in identifying the problem and it is on our future work list.

Looking at the results of MTFC-HA or MTFC-FA which are the contraflow schemes with the least number of zero-capacity edges, we also make an interesting observation. The results of Greedy and MTFC-HA are almost same except a few of runs, while MTFC-FA keeps the pace with MTFC-MZ in most runs. Even in the case when MTFC-HA has worse results than Greedy, the gap between the two results is in most cases much less than the gap between any MTFC results and Greedy results. However we don't find the peculiarly better results from MTFC-HA as opposed to MTFC-MZ. MTFC-FA generates better results even when MTFC-MZ results are worse as in 1-st of 200 nodes or 7-12/12-th of 300 nodes. However in some cases such as 12-th of 200 nodes, MTFC-FA generates the worst result. Careful investigation about those peculiar results will give us a lot of additional information and we are planning so.

VI. Conclusions

In this paper we defined the maximum throughput flow and presented an efficient and highly scalable algorithm for contraflow evacuation routing based on maximum throughput flows. The presented algorithm MTFC utilizes the reverse shortest path algorithm to reduce the path-finding time. Maximum throughput flow concept helps MTFC generate contraflow schemes that result in short evacuation time (to the equivalent level as the Greedy algorithm in [11]) and the reverse shortest path algorithm helps MTFC dramatically reduce running time to the level of maximal scalability. From the computation results, we made some interesting observation about the peculiarly good results of MTFlow in some runs and we are planning to further investigate the cases.

Our future plans also include careful comparison of the contraflow schemes using other evacuation routing algorithm. The thorough theoretical study of the proposed algorithm is also one of our future directions. For the full envisioning of the ad hoc contraflow scheme, distributed version of the proposed algorithm will be studied too.

References

[1] T.J. Cova and J.P. Johnson, *A network flow model for lane-based evacuation routing*. Transportation Research Part A, vol. 37, 2003, pp. 579–604.

[2] J. G. Doheny and J. L. Fraser, *Mobedic - a decision modeling tool for emergency situations*. Expert Systems With Applications, 10:17–27, 1996.

[3] M. Ebihara and 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.

[4] Y. Fan and Y. Xinping and X. Kun, *Evacuation Flow Assignment based on Improved MCMF Algorithm*. Proc. First International Conference on Intelligent Networks and Intelligent Systems, 2008, pp. 637–640.

[5] Florida Department of Transportation, *Florida's One-Way Evacuation Operation*. http://www.onewayflorida.org/, 2012.

[6] L.K. Fleischer, *Faster Algorithms For The Quickest Transshipment Problem*. SIAM J. Optim., vol. 12/1, 2001, pp. 18–35.

[7] G. Ford and R. Henk and P. Barricklow, *Interstate highway 37 reverse flow analysis - technical memorandum*. Technical report, Texas Transportation Institute, 2000.

[8] H.W. Hamacher and S.A. Tjandra, *Mathematical Modelling of Evacuation Problems: A State of Art*. Technical Report Nr. 24, Berichte des Fraunhofer ITWM, 2001.

[9] G. Hamza-Lup and K. A. Hua and 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.

[10] S. Kim and S. Shekhar, *Contraflow Network Reconfiguration for Evacuation Planning: A Summary of Results*. Proc. Proceedings of the 13th ACM Symposium on Advances in Geographic Information Systems, 2005, pp. 250–259.

[11] S. Kim and S. Shenkhar and M. Min, *Contraflow Transportation Network Reconfiguration for Evacuation Route Planning*. IEEE Transactions on Knowledge and Data Engineering, vol. 20/8, 2008, pp. 1115–1129.

[12] T. Litman *Lessons From Katrina and Rita: What Major Disasters Can*. Teach Transpor tation Planners. Journal of Transportation Engineering, 132(1):11–18, 2006.

[13] Q. Lu and Y. Huang and S. Shekhar, *Evacuation Planning: A Capacity Constrained Routing Approach*. Lecture Note on Computer Science, vol. 2665, 2003, pp. 111–125.

[14] Q. Lu and B. George and S. Shekhar, *Capacity Constrained Routing Algorithms for Evacuation Planning: A Summary of Results*. Proc. 9th International Symposium on Spatial and Temporal Databases, 2007, pp. 291–307.

[15] Q. Lu and B. George and S. Shekhar, *Evacuation Route Planning: Scalable Heuristics*. Proc. 15th International Symposium on Advances in Geographic Information Systems, 2007.

[16] M. Min and B.C. Neupane, *An Evacuation Planner Algorithm in Flat Time Graphs*. Proc. of ACM International Conference on Ubiquitous Information Management and Communication (ICUIMC), 2011.

[17] M. Min, *Synchronized Flow-Based Evacuation Route Planning*. Proc. of International Conference on Wireless Algorithms, Systems, and Applications , 2012.

[18] J. Orlin, *Maximum-throughput dynamic network flows*. Mathematical Programming, 27(2):214–231, 1983.

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

[20] 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.

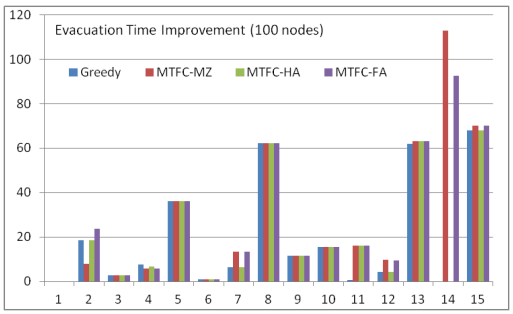
[21] H. Tuydes and A. Ziliaskopoulos, *Network Re-design to Optimize Evacuation Contraflow*. Technical Report 04-4715, Presented at 83rd Annual Meeting of the Transportation Research Board, 2004.

[22] H. Tuydes and A. Ziliaskopoulos, *Tabu-Based Heuristic for Optimization of Network Evacuation Contraflow*. Technical Report 06-2022, Presented at 85rd Annual Meeting of the Transportation Research Board, 2006.

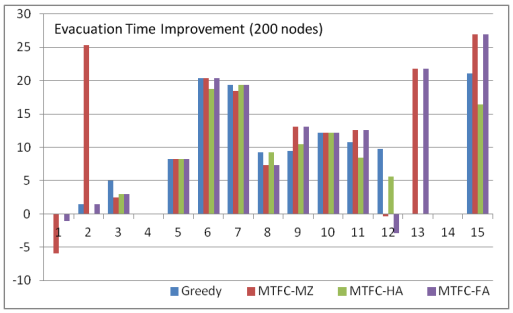
[23] B. Wolshon, *One-way-out: Contraflow freeway operation for hurricane evacuation*. Natural Hazards Review, 2(3):105–112, 2001.

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

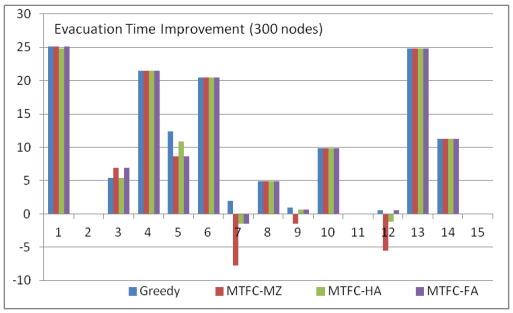
[25] D. Yin, *A Scalable Heuristic for Evacuation Planning in Large Road Network*. Proc. the Second International Workshop on Computational Transportation Science, 2009, pp. 19–24.



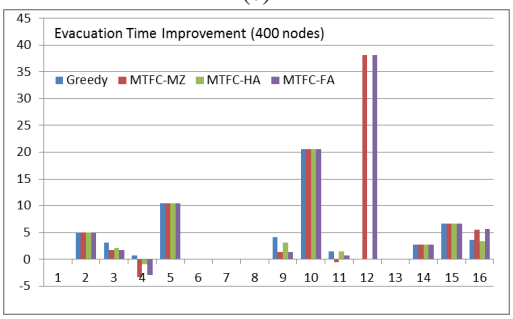
(a)



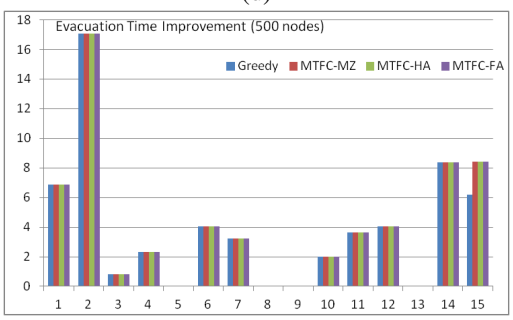
(b)



(c)



(d)



(e)

Fig. 5. Improvement of Evacuation Time Against Non-contraflow Evacuation Time