To appear in Proceedings of the 14th IEEE ITS Conference (ITSC 2011), Washington DC, USA, October 2011.

Dynamic Lane Reversal in Traffic Management

Matthew Hausknecht, Tsz-Chiu Au, Peter Stone
Department of Computer Science
University of Texas at Austin
{mhauskn,chiu,pstone}@cs.utexas.edu

David Fajardo, Travis Waller
School of Civil and Environmental Engineering
University of New South Wales
{davidfajardo2,s.travis.waller}@gmail.com

Abstract— **Contraflow lane reversal—the reversal of lanes in order to temporarily increase the capacity of congested roads—can effectively mitigate traffic congestion during rush hour and emergency evacuation. However, contraflow lane reversal deployed in several cities are designed for specific traffic patterns at specific hours, and do not adapt to fluctuations in actual traffic. Motivated by recent advances in autonomous vehicle technology, we propose a framework for *dynamic* lane reversal in which the lane directionality is updated quickly and automatically in response to instantaneous traffic conditions recorded by traffic sensors. We analyze the conditions under which dynamic lane reversal is effective and propose an integer linear programming formulation and a bi-level programming formulation to compute the optimal lane reversal configuration that maximizes the traffic flow. In our experiments, active contraflow increases network efficiency by 72%.**

I. INTRODUCTION

Traffic congestion is a major issue in today's transportation systems. Contraflow lane reversal, the reversal of traffic flow along a lane to temporarily increase the capacity of congested roads at the expense of under-utilized ones, is a method to increase traffic flow without adding additional roads or lanes. On the left of Fig. 1, the top lanes are being more heavily utilized than the bottom ones. On the right, by temporarily converting a lane to flow in the opposite direction, the instantaneous capacity in the left-to-right direction of the road is increased by 50%. Contraflow lane reversal has been used routinely in several cities in order to alleviate traffic during rush hours as well as to reroute traffic around certain areas such as construction sites or stadiums.

Today, contraflow lane reversal is used at a macro time scale at rush hour or for quick evacuations from an area. In both cases however, the change in flow must be carefully planned before the event, with little or no room for dynamic changes. Today's hardware for traffic monitoring is good enough to gather real-time traffic data. With the help of modern computerized traffic control systems, it is possible to quickly and dynamically open and close lanes or entire roads, or even change the directionality of lanes based on real-time usage statistics, such that effective capacity of a road can be dynamically changed based on the demand.

Rapid changes of lane directions, however, may confuse human drivers. To fully utilize the potential of dynamic lane reversal, we will need to rely on the upcoming availability of computer-aided driving systems and fully autonomous vehicles that will help vehicles to adjust to the rapid changes of lane directions. With the help of computerized driving systems, more aggressive contraflow lane reversal strategies can be implemented to improve traffic flow of a city without increasing the amount of land dedicated to transportation.

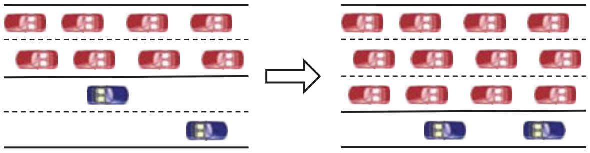

Fig. 1. An illustration of contraflow lane reversal (cars are driving on the right side of the road). The total capacity of the road is increased by approximately 50% by reversing the directionality of a middle lane.

An important component of implementing dynamic lane reversal is fully understanding the systemwide impact of increasing capacity on an individual link. We define the objective of contraflow as follows: given a road network, a specification of vehicles' locations and destinations, and a method for determining network efficiency (such as an objective function), assign a direction of flow to each lane such that network efficiency is maximized. To study the network effects of dynamically repurposing lanes, we cast the problem as a maximum multi-commodity flow problem—a version of the maximum flow problem in graph theory with multiple commodities (or goods) flowing through the network. Then we propose an integer programming formulation and a bi-level programming formulation to compute the maximum flow in the network. We evaluate our approaches in grid-like transportation networks representative of many downtown metropolitan areas where it will have the most potential impact.

The rest of the paper is organized as follows. In Section II, we discuss the hardware needed for implementing a dynamic lane reversal scheme. In Section III and IV, we analyze under what conditions dynamic lane reversal will be useful for an individual road and intersection. In Section V and VI, we introduce both the macroscopic ILP traffic model as well as the bi-level formulation, and investigate the performance gains imparted by dynamically reconfiguring lanes.

II. HARDWARE FOR DYNAMIC LANE REVERSAL

A reversible lane (or contraflow lane) is a lane in which traffic may travel in either direction. The common hardware for creating reversible lanes is overhead traffic lights (Fig. 2(a)). In many cities, barrier transfer machines, also known as zipper machine, are used to relocate the moveable barriers such that the road in one direction can be dynamically widened at the expense of the other (Fig. 2(b)).

(a) Signals of lane direction (b) Barrier transfer machines

Fig. 2. Hardware for controlling contraflow lane.

The basic requirements to support dynamic lane reversal are that the reversal has to be done quickly and safely, and that the drivers must be notified about the change immediately. While it's conceivable to devise a system that satisfies these requirements using the hardware in Fig. 2, there is likely to be significant cost and risk of driver confusion. However once most cars are controlled by computer, these costs and risks may be significantly reduced by real time updates of lane direction over wireless network communication, as computerized driving systems (i.e., autonomous vehicles) can react to the changes of lane directions much quicker than human drivers. For example, Dresner and Stone proposed an intersection control mechanism called Autonomous Intersection Management (AIM) that uses a wireless communication protocol to enable fine-grained interleaving of vehicle routes through an intersection [1]. With some modifications to the AIM protocol, autonomous vehicles can be informed about the current lane directions as well.

III. LANE REVERSAL ON ONE ROAD

We begin by considering, from a theoretical perspective, the effects of lane reversal on a *single* road. Consider a road between intersections I_1 and I_2. Let R be the road between I_1 and I_2, $L_{1,2}$ be the set of lanes from I_1 to I_2, and $L_{2,1}$ be the set of lanes from I_2 to I_1. As an example, in Fig. 3, $L_{1,2} = \{l_1, l_2\}$ and $L_{2,1} = \{l_3, l_4\}$. The capacity of a lane l, denoted by $c(l)$, is the maximum rate at which vehicles enter the lane and is measured by the number of vehicles per hour. We assume the capacity of a set L of lanes, denoted by $c(L)$, is the sum of the capacities of all lanes ($c(L) = \sum_{l \in L} c(l)$). For simplicity, we ignore the effect of lane changing which potentially reduces the capacity of L.

Assume both I_1 and I_2 are sources at which vehicles are "generated" to travel along R at the *target traffic rates* $\beta(I_1)$ and $\beta(I_2)$ respectively. But the *effective traffic rates* $\lambda(L_{1,2})$ and $\lambda(L_{2,1})$ at which vehicles actually enter the road are limited by the capacity of the lanes. More precisely, $\lambda(L_{1,2}) = \min\{\beta(I_1), c(L_{1,2})\}$ and $\lambda(L_{2,1}) = \min\{\beta(I_2), c(L_{2,1})\}$. If $\lambda(L_{1,2}) = c(\lambda(L_{1,2}))$, we say $L_{1,2}$ is *saturated*. If $\beta(I_1) > c(L_{1,2})$, $L_{1,2}$ is *oversaturated* by an amount of $\beta(I_1) - c(L_{1,2})$. Clearly, if $L_{1,2}$ is oversaturated, $L_{1,2}$ is saturated. $L_{1,2}$ is *undersaturated* by an amount of $c(L_{1,2}) - \beta(I_1)$ if $\beta(I_1) < c(L_{1,2})$. Clearly, if $L_{1,2}$ is undersaturated, $L_{1,2}$ is not saturated. The saturation of $L_{2,1}$ is defined in the same manner.

The *throughput* of the road R is the sum of the effective traffic rates of the lanes (i.e., $\lambda(L_{1,2}) + \lambda(L_{2,1})$). Now consider what happens if the direction of $l \in L_{1,2}$ is reversed.

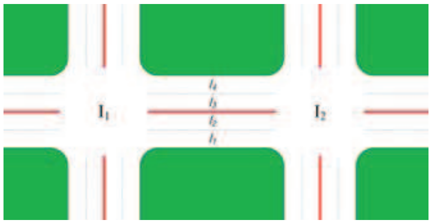

Fig. 3. Lane reversal in a road.

By definition, the throughput of the road increases after the reversal of l if and only if

$$\lambda(L_{1,2}) + \lambda(L_{2,1}) < \lambda'(L_{1,2}) + \lambda'(L_{2,1}) \qquad (1)$$

where $\lambda'(L_{1,2}) = \min\{\beta(I_1), c(L_{1,2}) - c(l)\}$, and $\lambda'(L_{2,1}) = \min\{\beta(I_2), c(L_{2,1}) + c(l)\}$.

In general, lane reversal is beneficial only when one of the directions is oversaturated while the other is undersaturated, as shown in Fig. 1. Formally, we have the following theorem:

Theorem 1: The throughput of the road R increases after the reversal of a lane $l \in L_{a,b}$ if and only if $L_{a,b}$ is undersaturated by δ_a while $L_{b,a}$ is oversaturated by δ_b, where $\max\{c(l) - \delta_a, 0.0\} < \delta_b$.

Proof Sketch. Due to space limitations, we only consider the case in which $c(l) > \delta_a$ and $\delta_b < c(l)$. The reversal of l reduces the effective traffic rate of $L_{b,a}$ by $x = c(l) - \delta_a > 0$ while the effective traffic rate of $L_{a,b}$ increases by δ_b. Thus the throughput of the road increases if and only if $\delta_b > x = \max\{c(l) - \delta_a, 0.0\}$. □

IV. LANE REVERSAL FOR AN INTERSECTION

Analyzing the change of the intersection throughput is necessary because intersections may potentially be the bottlenecks of the traffic flow, preventing the adjacent roads from achieving their maximum throughput as predicted by Theorem 1. Estimating the effects on intersection throughput theoretically, however, can be a challenging task, especially when vehicles from different roads can enter the intersection at the same time. Therefore, we use empirical methods to see whether intersections can handle the increase of the incoming traffic when the directions of adjacent lanes reverse.

We experiment with the intersection in Fig. 4, which has six lanes on each incident road. Initially, 3 lanes are incoming lanes and 3 lanes are outgoing lanes. We set the target traffic rate of the eastbound road be 5500 vehicles per hour, the target traffic rate of westbound road be 1100 vehicles per hour, and the traget traffic rates of both northbound and southbound roads are 1650 vehicles per hour. Thus, the traffic on the eastbound road is several times higher than other roads, causing traffic congestion on the eastbound road. We check whether reversing the direction of two lanes on the westbound road can help to increase the throughput of the eastbound road as well as the intersection throughput (the number of vehicles entering the intersection per hour). The new lane configuration is shown on the right side in Fig. 4.

We repeated the experiment 30 times and in each run we measured 1) the total number of vehicles entering the intersection during the 1-hour period, and 2) the number of

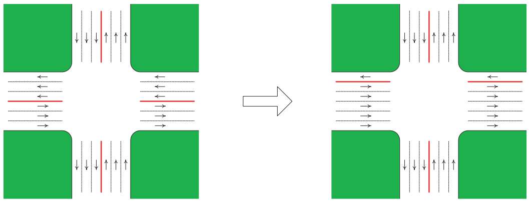

Fig. 4. The reversal of two lanes on the westbound road of an intersection.

TABLE I

	Before reversal	After reversal	Change
Eastbound road	4618.5 ± 8.8	5228.0 ± 12.2	13.2%
Westbound road	1184.5 ± 14.1	1124.7 ± 9.6	−5.0%
Northbound road	1711.6 ± 11.3	1700.6 ± 13.6	−0.6%
Southbound road	1712.8 ± 13.8	1714.6 ± 12.1	−0.1%
Intersection	9183.3 ± 32.4	9775.8 ± 26.6	6.5%

vehicles entering the intersection from each road during a 1-hour period. The average of the number of vehicles and the 95% confidence intervals are shown in Table I. The results show that the throughput of the intersection increased by 6%, and this is mainly due to the increase of incoming traffic from the eastbound road whose throughput is increased by 13%. Note that both increases are statistically significant. After lane reversal, the eastbound road's traffic rate is much closer to the target traffic rate, and this means that the intersection successfully handled the increase of the traffic coming from the eastbound road. The lane reversal has only minor detrimental effects on other roads, because they are undersaturated and the lane reversal does not reduce the capacity of these roads below their target traffic rate.

V. ADAPTIVE CAPACITY FOR A ROAD NETWORK

While the ability to improve throughput on individual roads and at individual intersections are important proofs of concept, the true question is whether (and how much) dynamic lane reversal can help on a full road network. To address this question we model a road network as a graph consisting of vertices and edges. Each node of the graph represents an intersection and each edge represents a road between intersections. Additionally, each u,v edge has an associated capacity $c(u,v)$ which constrains the maximum amount of traffic that road can handle (in this section we model intersections as having infinite capacity).

Traffic in the network is modeled in terms of aggregate demand. Specifically we consider a finite number of flows, where each flow has an associated source vertex, destination vertex, and integer-valued demand. For example, a parking garage at a mall could be a source, and a bridge at the edge of the network could be a destination. The demand represents the instantaneous number of vehicles that want to travel between these two points. Vehicles may take any path between the source and destination so long as they do not violate capacity constraints of roads. We seek to determine 1) whether or not the lanes of a given road network can be dynamically reconfigured in order to accommodate a given set of traffic flows and 2) what is the maximum demand a network can handle.

A. Multicommodity Flow Problem

The multi-commodity flow problem is a generalization of the well-known max flow problem in which multiple commodities or goods flow through the network, each with different source and sink nodes. Modeling a road network at the macroscopic level allows us to map the well-studied problem of multi-commodity flow directly onto our problem of dynamically reconfiguring lanes. In order to solve this problem we utilize the the mathematical machinery of linear programming.

B. ILP Formulation

A Linear Program contains a linear function to be maximized over a set of variables, subject to constraints. In normal linear programs these variables are allowed to assume fractional values, but since all of our flow demands are required to be integer-valued, we must approach this problem as an Integer Linear Program. Unfortunately, the multicommodity flow problem has long been known to be NP-complete when dealing with integer flows, even for only two commodities [2].

We define the following Integer Linear Program: Given a graph $G = \{V,E\}$, each edge (u,v) has some integer capacity $c(u,v)$ representing the total number of lanes present on that road. There are k distinct commodities (traffic flows) $K_1,\ldots,K_k$ where each commodity $K_i = (s_i,t_i,d_i)$ has an associated source s_i, destination t_i, and demand d_i. Flow of commodity i over edge (u,v) is denoted $f_i(u,v)$.

Our objective is to find an assignment of flows which satisfy the following three constraints: The capacity constraint, shown in Equation 2, specifies that the total amount of flow (in both directions) over a given (u,v) edge must not exceed the capacity of that edge (note $c(u,v) = c(v,u)$ in the undirected case). The conservation constraint, Equation 3, ensures that for all non sink/source vertices the amount of inflow of a given commodity equals the amount of outflow. Finally Equation 4 specifies that the flow of each commodity must meet or exceed the demand for that commodity.

$$\forall_{(u,v)\in E}\sum_{i=1}^{k}(f_i(u,v)+f_i(v,u)) \leq c(u,v) \tag{2}$$

$$\forall_{i\in 1\ldots k, v\in V-\{s,t\}}[\sum_{u\in V} f_i(u,v) = \sum_{w\in V} f_i(v,w)] \tag{3}$$

$$\forall_{i\in 1\ldots k}\sum_{w\in V} f_i(s_i,w) = \sum_{w\in V} f_i(w,t_i) \geq d_i \tag{4}$$

The goal of the ILP solver is to find an assignment of directionality to each lane which maximizes the objective function (Equation 5) subject to the constraints specified above. This is done by assigning integer values to individual flows. We choose to use the maximum multi-commodity objective function in which the objective is to reconfigure the network to maximize the sum of all commodity flows:

$$maximize \sum_{i=1}^{k}\sum_{w\in V} f_i(s_i,w) \tag{5}$$

While multiple possible objective functions could meet our criterion of finding a lane configuration capable of handling a given set of flows, we choose the maximum multicommodity objective because it forces the ILP solver not only to satisfy the flow demands, but also to find the absolute maximum amount of traffic a network can handle. In contrast, the following section explores an alternative, least-cost objective function.

C. Bi-Level Programming Formulation

The multicommodity flow formulation of our problem is convenient in that its solution—the maximum flow—is unique and independent of vehicles' behavior. However it ignores the fact that drivers are self-interested—they are concerned about their own travel times and have no incentive to cooperate to achieve the maximum flow of the network. Therefore, we consider an alternative formulation for the adaptive capacity problem using a bi-level approach, where the objective is to set link capacities such that, as flows are determined by User Equilibrium behavior, the total system travel time, i.e. the sum of the travel times of all users, is minimized. By modeling the route choice user behavior as a travel cost minimization, we can more accurately characterize the behavior of users in a traffic context. The mathematical formulation is shown in Equations 6–8. The upper level problem includes the allocation of capacity x to each of the links, while the lower level problem is the classic User Equilibrium model presented by Wardrop [3]:

$$\min_x f(x) \quad \text{s.t.} \quad -c_{ij} < x_{ij} < c_{ji} \qquad \forall i < j \quad \text{where} \tag{6}$$

$$f(x) = \min_v \sum_{(i,j)} \int_0^{\sum_{od} v_{ij}^{od}} t_f \left(1 + \alpha_{ij} \left(\frac{w}{c_{ij} + x_{ij}} \right)^{\beta_{ij}} \right) dw \tag{7}$$

$$\text{s.t.} \quad \sum_j v_{ij}^{od} - \sum_j v_{ji}^{od} = b^{od}(i) \qquad \forall i, od. v_{ij} \geq 0 \qquad \forall i, j \tag{8}$$

where c is the capacity vector, v^{od} is the flow vector for each OD, x is the dynamic lane capacity allocation vector, t_f is the free flow speed vector, and α and β are parameters, and $b^{od}(i)$ is equal to the node supply/demand for each OD pair od. As bi-level problems such as this are difficult to solve exactly, we present a Genetic Algorithms based solution approach in the following section.

D. Genetic Algorithm Solution Method

Genetic algorithms (GAs) is a global search heuristic that uses techniques inspired by evolutionary biology ([4], [5]). GAs are based on the assumption that the best solution is found in regions of solution space having a high proportion of good solutions. GAs explore the solution domain to identify the promising region and then search the promising regions more intensely. GAs start with a population of randomly regenerated individuals that evolves with generations based on the principle of survival of the fittest. Unlike classical methods, GAs work with a population of points. Therefore, the chances of getting trapped at local optima are reduced. Moreover, many variations of GAs are suitable for handling complex problems involving discontinuities, disjoint feasible spaces, and noisy function evaluation [6], [7], [8]. In our formulation, a gene represents the capacity allocated to a link (i,j) that is taken from the reverse link (j,i). The fitness function used is the total system travel time in the underlying UE problem given a decision vector representing capacity allocation.

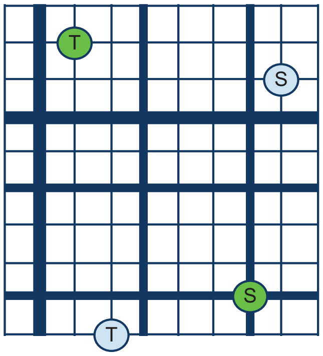

Fig. 5. A example generated graph with two incident S,T flows. Thick lines represents highways, medium lines represent arterial roads, and thin lines represent streets.

TABLE II

ROAD TYPES COMPOSING THE RANDOMLY GENERATED NETWORKS.

Road Type	Capacity Range	Probability
Street	1-4	1.0
Arterial Road	6-15	.1
Highway	24-32	.01

VI. ILP EXPERIMENTS

In this section we use the ILP solver to empirically compare different traffic management systems – those which can reverse lane directions quickly and those which can reverse slowly or not at all. We hypothesize that traffic managers which have the ability to quickly reverse the flow of traffic along lanes will achieve higher throughputs than slower traffic managers.

A. Generated Graphs

We automatically generate graphs qualitatively similar to downtown regions of many cities. Each graph takes the form of a connected planar grid. To determine the capacities for each road, we randomly select from one of the three road types shown in Table II with the associated probabilities. Flows are generated by selecting a random source, sink vertex pair from the graph. An example road network is shown in Figure 5.

B. Experimental Evaluation

To measure the performance difference between traffic control systems with the ability to quickly reverse lanes (such as the AIM protocol) and those that can only slowly reverse lanes (such as zipper machines), we evaluate each traffic management system for 10 hours. Each hour a new set of random flows is spawned, and it is the job of the traffic manager to accommodate this traffic as well as possible by reversing lane directionality. However not all traffic managers will be able to reverse lanes every hour. Traffic management systems differ in their *reconfiguration period* – the amount of time that must elapse between lane reconfigurations. For example, a system with reconfiguration period of 2 will reconfigure lane directions every other hour while a system with reconfiguration period of 5 will only reconfigure every fifth hour. If a traffic manager is unable to reconfigure lanes for a given hour, the network throughput is computed by finding the maximum multicommodity flow over the current lane configuration (e.g. directed multicommodity flow) without allowing the directions of any lanes to change. On the other hand, if a traffic manager is able to reconfigure lanes for a given hour, we compute throughput by finding the maximum multicommodity flow over the network consisting entirely of undirected edges. This gives managers with low reconfiguration period the ability to more fully adapt to changing traffic conditions.

An example run proceeds as follows: we first randomly generate a road network using the procedure described above. Initially this network is configured in a balanced manner in which the capacity of each road is divided as evenly as possible between lanes flowing in one direction and those in the opposite. Next, random flows are generated and the ILP solver is used to compute the network throughput for that hour either on the current directed configuration (if the manager cannot reconfigure this hour) or on the undirected configuration (if the manager can reconfigure this hour). Any changes made to the directionality of lanes carry over into the next hour when new flows are generated.

In our experiments we evaluated networks of size 100 (10x10) for 10 hours. Each hour contained a set of 4 randomly generated flows. Demands for these flows were set to 0 to ensure that the ILP solver would both reach a valid solution and maximize the achievable throughput.[1] The throughput of each traffic manager was evaluated at each timestep and the total throughput for a traffic manager with a given reconfiguration period was the sum of the throughputs it achieved over all 10 hours. We evaluated the performance of each traffic manager over 34 different networks (34 trials) for a total of 340 total timesteps.

Figure 6 shows the total network throughput achieved by traffic managers with different reconfiguration periods. The results show a significant increase in performance of traffic managers who have some form of lane reconfiguration in comparison to no lane reconfiguration. This is not surprising since the benefits of contraflow are well established. However, we seek to address the question of how much performance increase is bestowed by frequent rather than infrequent lane reconfiguration. Infrequent reconfiguration (*reconfiguration period* = 3,4,5) shows only modest improvements over the static configuration, approximately 11% throughput gain. However, decreasing our reconfiguration period to 2, we see a 32% performance gain over the static case. Finally, the fully dynamic reconfiguration period 1 traffic manager provides a 72% increase in throughput compared to the static network. This trend suggests that the large gains in traffic efficiency are achievable with reconfiguration technology which can adapt quickly to changing traffic flows and quickly reconfigure lanes.

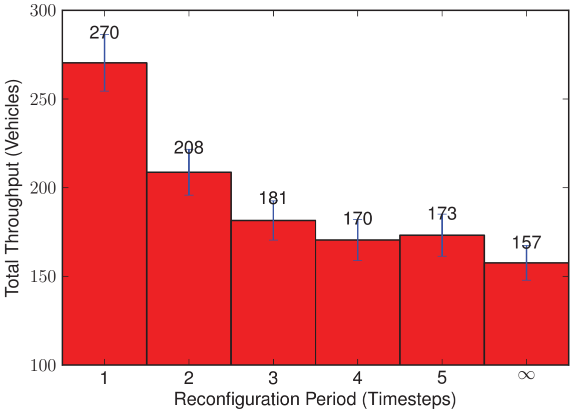

Fig. 6. Total network throughput (vehicles) as a function of DLR period. A reconfiguration period of 1 means that the network was reconfigured each timestep; 2 means every other timestep; ∞ means the network was never reconfigured (always in balanced, directed configuration). Error bars denote 95% confidence intervals. Period 1,2,3 are statistically significant with respect to period 0 as well as each other.

VII. BI-LEVEL FORMULATION EXPERIMENTS

This section shows preliminary results of the implementation of the bi-level formulation presented in Section V-C. The objective of these numerical results is to show that the algorithm is implementable for modest sized problems.

We solve a problem on a 10x10 grid network, where the number of lanes varies for each link, the capacity is 1800 vehicles/hour/lane, $\alpha = 0.1$, $\beta = 4$, and t_f is 25 mph, 35 mph and 55 mph depending on the number of available lanes. For the GA, the size of the population and the number of generations were both set to 30, the mutation probability was set to 0.002, and the cross-over probability was set to 0.75.

Based on these parameters, the model was solved for each of 10 time periods. In order to compare the behavior of the GA based solution method with the ILP solver, we examine the resulting lane configurations after each time period and calculate the correlation coefficient of the solution vectors for the ILP and GA. The correlations in lane configuration for each of the 10 hours are shown in Table III. As shown by the resulting correlation coefficients, the two solution vectors are quite different, which conforms to the idea that increased accountability in the driver route choice response process will lead to significantly different solution strategies.

While the bi-level and ILP solutions show significant differences, we expect that both solutions are valid under the different assumptions each model makes. Because of the highly redundant grid based network topology, we expect that (source, destination) demands may be satisfied by multiple possible paths. We hypothesize that the discrepancy in the solutions results from different paths being utilized for the same set of demands. However, while both models are finding valid solutions given their constraints, we should expect the bi-level formulation's solution to be more realistic in an actual traffic network because the bi-level model incorporates many aspects of real traffic such as congestion, equilibrium

---

[1] Incorporating non-zero demands is straightforward to implement but could have resulted in the ILP solver being unable to find valid solutions in some cases.

TABLE III
CORRELATION COEFFICIENTS OF ILP AND GA SOLUTION VECTORS

Hour	1	2	3	4	5	6	7	8	9	10
Corr.	0.1	0.2	0.1	0.3	-0.1	0.3	0.3	0.3	0.6	0.1

and speed limits. On the other hand, the bi-level model can guarantee only approximate solutions while the ILP solver computes the optimal solution.

VIII. RELATED WORK

Contraflow has been studied extensively in scenarios involving emergency evacuations from specific locations [11], [12], [12], [13], [14]. For example, early work by Zhou et al. [9] and Xue [10] developed contraflow lane control systems for managing the George Massey tunnel. These systems predicted traffic demand and reversed lanes in order to reduce congestion and delays. In contrast, our approach is applicable to any road network which can be represented as graph. Heuristic solutions to contraflow for emergency evacuations were considered by several authors [15], [16], [17]. Particularly, Kim et al. [18] examine evacuation scenarios with multiple sources. However, their model does not allow individual lanes to be reversed and instead requires that each road be converted into a single direction of flow. Cova and Johnson [19] present a model for identifying optimal lane-based evacuation routing plans for road networks using a mixed integer programming solver. Unlike most models, they consider the effects of intersection merging on overall traffic and seek to find routing solutions which avoid crossing flows of traffic. However, this work considers only evacuations on static road networks and does not attempt to reverse or reconfigure lanes. In contrast to these authors, we do not take the perspective of an evacuation and instead focus on contraflow to optimize every-day traffic efficiency.

IX. CONCLUSIONS AND FUTURE WORK

In this paper, we propose a framework of dynamic lane reversal—directionality of lanes are updated quickly and automatically in response to instantaneous traffic conditions recorded by traffic sensors. Recent advances in robotic research leads us to believe that cars will be the key to unlocking the full potential of dynamic lane reversal schemes since they eliminate human errors. Here we established theoretical conditions under which contraflow lane reversal can increase the efficiency of a road, and showed empirically that contraflow has beneficial effects on intersection throughput. We then formulated the problem of contraflow lane reversal as a multicommodity flow problem, and described how to compute the optimal direction for each lane by an integer linear programming solver. Our experimental results indicate that a) contraflow in any form provides increased network efficiency and b) dynamic (fast-paced) lane reversals provide large performance gains (72% in our case) over slower pace lane reversal schemes. We also introduced a bi-level programming formulation which accounts for link level congestion and user equilibrium conditions, and compared the results generated by Genetic Algorithms to solutions in the multicommodity flow formulation. Our ongoing research agenda includes designing car control and intersection control policies for dynamic lane reversal.

Acknowledgments. This work has taken place in part in the Learning Agents Research Group (LARG) at UT Austin. LARG research is supported in part by NSF (IIS-0917122), ONR (N00014-09-1-0658), and the FHWA (DTFH61-07-H-00030).

REFERENCES

[1] K. Dresner and P. Stone, "A multiagent approach to autonomous intersection management," *Journal of Artificial Intelligence Research (JAIR)*, March 2008.

[2] S. Even, A. Itai, and A. Shamir, "On the complexity of timetable and multicommodity flow problems," *SIAM J. Comput.*, vol. 5, no. 4, pp. 691–703, 1976.

[3] J. Wardrop, "Road paper. some theoretical aspects of road traffic research." in *ICE Proceedings: Engineering Divisions*, vol. 1, no. 3. Ice Virtual Library, 1952, pp. 325–362.

[4] H. Adeli and S. Hung, *Machine learning: neural networks, genetic algorithms, and fuzzy systems*. John Wiley & Sons, Inc., 1994.

[5] H. Adeli and S. Kumar, "Distributed genetic algorithm for structural optimization," *Journal of Aerospace Engineering*, vol. 8, no. 3, pp. 156–163, 1995.

[6] C. Fonseca and P. Fleming, "An overview of evolutionary algorithms in multiobjective optimization," *Evolutionary computation*, vol. 3, no. 1, pp. 1–16, 1995.

[7] K. Sarma and H. Adeli, "Fuzzy genetic algorithm for optimization of steel structures," *Journal of Structural Engineering*, vol. 126, 2000.

[8] ——, "Fuzzy discrete multicriteria cost optimization of steel structures," *Journal of Structural Engineering*, vol. 126, p. 1339, 2000.

[9] W. Zhou, P. Livolsi, E. Miska, H. Zhang, J. Wu, and D. Yang, "An intelligent traffic responsive contraflow lane control system," in *Proceedings of the IEEE ehicle Navigation and Information Systems Conference*, 1993, pp. 174–181.

[10] D. Xue and Z. Dong, "An intelligent contraflow control method for real-time optimal traffic scheduling using artificial neural network, fuzzy pattern recognition, and optimization," *IEEE Transactions on Control Systems technology*, vol. 8, no. 1, pp. 183–191, 2000.

[11] M. Jha, K. Moore, and B. Pashaie, "Emergency evacuation planning with microscopic traffic simulation," *Transportation Research Record: Journal of the Transportation Research Board*, no. 1886, pp. 40–48, 2004.

[12] H. Tuydes and A. Ziliaskopoulos, "Network re-design to optimize evacuation contraflow," Proc. 83rd Annual Meeting of the Transportation Research Board, Tech. Rep. 04-4715, 2004.

[13] G. Theodoulou, "Contraflow evacuation on the westbound i-10 out of the city of new orleans," Master's thesis, Louisiana State University, 2003.

[14] G. Theodoulou and B. Wolshon, "Alternative methods to increase the effectiveness of freeway contraflow evacuation," *Transportation Research Record: Journal of the Transportation Research Board*, no. 1865, pp. 48–56, 2004.

[15] Q. Lu, B. George, and S. Shekhar, "Capacity constrained routing algorithms for evacuation planning: A summary of results," in *Proc. of 9th International Symposium on Spatial and Temporal Databases (SSTD'05)*, 2005, pp. 22–24.

[16] S. Kim and S. Shekhar, "Contraflow network reconfiguration for evacuation planning: A summary of results," in *Proc. 13th ACM International Symposium on Advances in Geographic Information Systems (GIS'05)*, 2005, pp. 250–259.

[17] B. Wolshon, "One-way-out: Contraflow freeway operation for hurricane evacuation," *Natural Hazards Review*, vol. 2, no. 3, pp. 105–112, 2001.

[18] S. Kim, S. Shekhar, and M. Min, "Contraflow transportation network reconfiguration for evacuation route planning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 8, pp. 1115–1129, 2008.

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