INTERNATIONAL JOURNAL OF CONTROL

Distributed dynamic lane reversal and rerouting for traffic delay reduction

Evan Gravelle[1] and Sonia Martínez[1]

[1]Department of Mechanical and Aerospace Engineering, University of California, San Diego, CA 92093, USA. {egravell,soniamd}@ucsd.edu

ARTICLE HISTORY
Compiled June 17, 2017

ABSTRACT
Traffic congestion is a major source of delays in modern road networks. Motivated by this, we propose two distributed algorithms to reduce delays: a dynamic lane reversal algorithm and a rerouting algorithm. When there is a density imbalance on a road, time can be saved by reallocating lanes from the less dense side to the more dense side, which motivates dynamic lane reversal. When a road has greater density than nearby roads, time can be saved by redirecting flow into the least congested roads, this motivates dynamic rerouting. Given a communication system between infrastructure and vehicles on the road, the local state of the network can be approximated and utilized by the algorithms to minimize travel time. In order to provide a better fundamental understanding of the system dynamics, we analyze equilibrium conditions for the system and prove convergence of the lane reversal algorithm to a critical point. Overall performance is also examined in simulation.

KEYWORDS
Traffic flow, traffic routing, lane reversal, distributed control.

1. Introduction

Motivation. Congestion is a major source of traffic delays in modern road networks, but the problem can be mitigated by smarter traffic systems. Significant imbalances of traffic density in a given road network can arise due to many events, such as when there is a large flow of vehicles towards an industrial center in the morning, a large event ends and there is a mass of flow out from large event, or there is an accident which creates heavy congestion on one side of a road. Modern infrastructure endowed with new information technology requires no additional space or construction and can substantially reduce overall traffic delays. Motivated by this, here we investigate the implementation and benefits of lane reversal and traffic rerouting distributed algorithms that can improve traffic flow.

In particular, recent advances in design, performance, and cost of autonomous vehicles (see Campbell, Egersdedt, How, and Murray (2010)) has fueled a growing interest

This work was supported by NSF-CMMI 1434819.

in Autonomous Intersection Management (AIM), an efficient policy for coordinating autonomous vehicles using an intersection manager (IM) to safely pass through an intersection Dresner and Stone (2008). With the help of the AIM policy and vehicle-to-infrastructure communications, an approximation of the state of traffic can be constructed. The IM can then implement more dynamic procedures to reverse one or more lanes or communicate a new route to some vehicles if traffic delays will be reduced. The future presence of autonomous vehicles is also important in implementing the actual lane reversal and vehicle rerouting, as physically moving a barrier to reverse a lane is a slow process that can take hours, Downey (2008), yet merely indicating a lane's direction or a new route for a vehicle is likely to cause driver confusion and increase risk of accident. With advances in vehicle autonomy, lane reversal and rerouting are less restricted by physical safety considerations and can be achieved through simple communication from the traffic signal to the vehicle.

Literature review. Many recent papers have furthered Autonomous Intersection Management. Batch processing of reservations in AIM to enforce liveness is proposed in Au, Shahidi, and Stone (2011). An auction-based scheme under AIM is analyzed in Carlino, Boyles, and Stone (2013). Local information is shared and utilized to minimize delay time under Greenshield's traffic model in Wuthishuwong and Traechtler (2013). Some effort has also recently been put towards solving vehicle routing problems in modern context. A provably safe distributed solution for coordinating vehicles outside an intersection is provided in Tallapragada and Cortés (2015). Work has also been done analyzing traffic evolution over networks. Classical traffic models are examined in a network setting in Work, Blandin, Tossavainen, and Bayen (2010). Passivity is used to generalize the network flow control problem in Wen and Arcak (2004). A solution to the problem of assigning freight loads to available carriers given unbalanced network conditions is found in Abadi, Ioannou, and Dessouky (2015).

Much of the literature concerning lane reversal discusses evacuation procedures in order to respond effectively to natural disasters, Chiu, Zheng, Villalobos, Peacock, and Henk (2008), Wang, Wang, Zhang, Ip, and Furata (2012). These papers propose the solution of lane reversal to accommodate emergency evacuation in a non-dynamic way. Some works discuss procedures and results for location-specific cases where lane reversal would be beneficial, Zhou, Livolsi, Miska, Zhang, Wu, and Yang (1993), Xue and Dong (2000). More recently, some have attempted to further improve results through dynamic lane reversal. The solution presented in Hausknecht, Au, and Stone (2011b) requires a centralized computer to find an allocation strategy, with a minimum timestep of one hour. In Meng, Khoo, and Cheu (2014), the authors formulate a model and present a centralized solution which does not use network dynamics. In Hausknecht, Au, and Stone (2011a), dynamic lane reversal is implemented in limited form on a single road and tested in simulation.

Statement of contributions. In this paper we extend the cell transmission model to characterize the evolution of vehicle density in a road network and the effect of both lane reversal and rerouting on these dynamics. We establish objective functions with the goal of minimizing total vehicle time spent on the road, and propose two algorithms. Using lane reversal, we propose a distributed *dynamic* algorithm to efficiently calculate and implement an appropriate lane allocation and prove convergence of the algorithm to a more efficient solution. An important aspect of this work is to provide a fundamental understanding of the system dynamics. To do so, we analyze the long term behavior of a road network with balanced lanes, and establish its convergence to an equilibrium under certain regularity conditions on its sources and sinks. We also propose a distributed rerouting algorithm to more efficiently achieve this long-term equilibrium. We show through simulations performance gains using lane reversal and rerouting on various initial conditions. This work is an extension of Gravelle and Martínez (2016), containing the full proof of the main stability theorem, an additional remark concerning noise in the dynamics, and additional simulation figures and discussion.

Notation and Nomenclature. The set of real numbers (positive real numbers, integers) is denoted by $\mathbb{R}$ ($\mathbb{Z}$, resp.). Similarly, $\mathbb{R}^n$ ($\mathbb{Z}^n$) denotes the product space of n copies of $\mathbb{R}$ ($\mathbb{Z}$, resp.). The vector of ones with length n is denoted by $\mathbf{1}_n$. A *directed graph* G consists of a set of *vertices* V and a set of *directed edges* E, $G = (V, E)$, such that $E \subset V \times V$. Vertex a is an *out-neighbor* of vertex b if $(b, a) \in E$. Similarly, a is an *in-neighbor* of b if $(a, b) \in E$. Vertex a is a neighbor of b if b is an out-neighbor or in-neighbor of a. The set of out-neighbors (resp. in-neighbors) of a is denoted $\mathcal{N}_a^{\text{out}}$ (resp. $\mathcal{N}_a^{\text{in}}$). Matrix $A = \{a_{ij}\}$ satisfies $A \in \text{sparse}(G)$, for $G = (V, E)$, if $a_{ij} = 0$ when $(i, j) \notin E$. Given a vertex set V, V_r denotes the set of cells contained on road r.

2. Problem Statement

We consider traffic evolving over a road network. Each *road* consists of one or two *sides* for each direction of traffic flow and which have a given number of lanes. In addition, each side is divided into *cells* of length L, which are used to describe the evolution of traffic density, see Figure 1.

We define a directed graph $G_{\mathrm{C}} = (C, E_{\mathrm{C}})$ of cells $i \in C$, such that $(j, h) \in E_{\mathrm{C}}$ if traffic can flow from cell j to cell h. A side is defined as the set of connected cells bounded by a source, sink, or an intersection manager (IM). A source (resp. sink) is a special cell in which traffic only flows out (resp. flows in), while an IM is an intelligent traffic management system at an intersection of roads. The set of all roads is denoted by R and the set of neighbors of road r is denoted by $\mathcal{N}_r$, where two roads are neighbors if they share an intersection. We denote S as the set of all sides, $p, -p \in S$ are the two sides of a road, and $n = |C|$. The intersection graph $G_{\mathrm{Z}} = (Z, E_{\mathrm{Z}})$ consists of the vertex set Z containing all IMs and edges $(z_1, z_2) \in Z$ if there is exactly one road connecting intersections z_1 and $z_2 \in Z$. A cell which flows into a sink is contained in set $\underline{B}$ and a cell which receives flow from a source is contained in set $\overline{B}$.

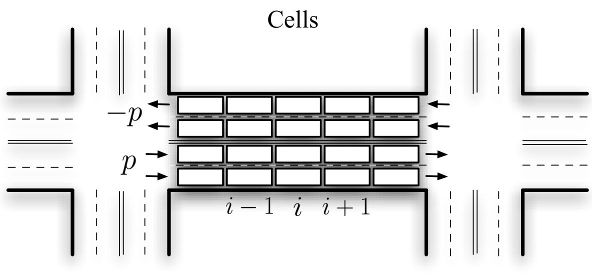

Figure 1. Road divided into cells

2.1. *Traffic Model*

The following traffic model is based on the *Lighthill-Whitham-Richards Partial Differential Equation*, Lighthill and Whitham (1955) and Richards (1956), to describe the evolution of vehicle density $\rho \in \mathbb{R}$ on each side,

$$\partial_t \rho + \partial_x Q(\rho) = 0. \tag{1}$$

This equation maintains conservation of mass, and the flow function $Q(\rho)$ is given by

$$Q(\rho) = \begin{cases} v_f \rho, & \rho \leq \rho_c, \\ \frac{v_f \rho_c}{\rho_{\text{jam}} - \rho_c}(\rho_{\text{jam}} - \rho), & \rho > \rho_c, \end{cases} \tag{2}$$

where $\rho \leq \rho_c$ is the condition for free flow, $\rho > \rho_c$ is the condition for congested flow, v_f is the free flow speed of the vehicles, ρ_{jam} is the density at which a traffic jam occurs, and ρ_c is the critical density value where maximum flow occurs, see Figure 2. This model is based on experimental data and is commonly used to model traffic flow, particularly because it is a simple model that captures the wave behavior of traffic.

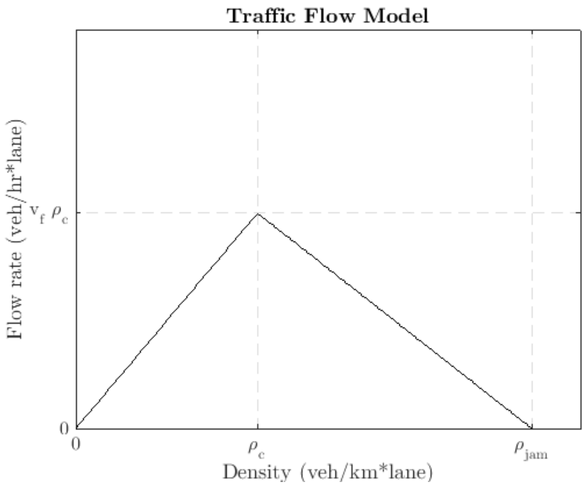

Figure 2. Vehicle flow model

The *Cell Transmission Model* Daganzo (1994) is a discretization of (1) using time step Δt and spatial step Δx, where is assumed that all cells have length $L = \Delta x$. For convenience, k indexes the discrete time step, with $t = t_0 + k\Delta t$. For a cell i with exactly one in-neighbor $i - 1$ and one out-neighbor $i + 1$, the density of the cell is updated according to

$$\rho_i(k+1) = \rho_i(k) + \frac{\Delta t}{L(\ell_p + u_p)}(q_{i-1,i}(k) - q_{i,i+1}(k)),$$

where i is contained on side p of road r, ℓ_p is the number of default lanes of side p, $\rho_i(k)$ is the density (veh/lane-km) of vehicles on i at time k, $q_{a,b}$ is the flow rate (veh/hr) from cell a to cell b, and $u_p(k) \in \{1-\ell_p, \dots, \ell_{-p}-1\}$ is the number of additional lanes on side p. The constraint $u_p + u_{-p} = \ell_p + \ell_{-p}$ must hold to keep the total number of lanes in a road constant, where $u_{-p} \in \{\ell_{-p}-1, \dots, \ell_p - 1\}$ is the number of additional lanes on side $-p$. We have

$$q_{i-1,i}(k) = \min\{\mathbf{q}_{i-1}(k), \mathbf{q}_i(k)\}, \tag{3}$$

with the piecewise functions $\mathbf{q}_{i-1}(k)$ and $\mathbf{q}_i(k)$ defined as

$$\mathbf{q}_{i-1}(k) = \begin{cases} v_f(\ell_p + u_p(k))\rho_{i-1}(k), & \rho_{i-1}(k) \le \rho_c, \\ v_f(\ell_p + u_p(k))\rho_c, & \rho_{i-1}(k) > \rho_c, \end{cases}$$
$$\mathbf{q}_i(k) = \begin{cases} v_f(\ell_p + u_p(k))\rho_c, & \rho_i(k) \le \rho_c, \\ \frac{v_f \rho_c}{\rho_{\text{jam}} - \rho_c}(\ell_p + u_p(k))(\rho_{\text{jam}} - \rho_i(k)), & \rho_i(k) > \rho_c. \end{cases}$$

Intuitively, the flow from $i-1$ to i is restricted when $\rho_{i-1}(k)$ is small or $\rho_i(k)$ is large Bretti, Natalini, and Piccoli (2006).

Cells can also be connected to sources or sinks of various strengths, these make up the boundary to the system. A source or sink is just like another cell but with an *effective density* given by

$$\begin{aligned} \rho_i &= \alpha_i \rho_c, & i \text{ is a source}, \\ \rho_i &= (1-\beta_i)\rho_c, & i \text{ is a sink}, \end{aligned}$$

where $\alpha_i(\beta_i)$ is the strength of the source (sink), resp.

To model a network of roads at intersections, the flow out of a cell must equal the sum of flows into other cells. We define a matrix $K = \{k_{ij}\} \in \mathbb{R}^{n \times n}$ where k_{ij} contains the fraction of vehicles which move from cell i to cell j. For now, we assume K is constant. If j is the only out-neighbor of i in G_{C} then $k_{ij} = 1$, but if j is one of multiple out-neighbors, then $k_{ij} < 1$. The flow out of any cell i, based on conservation of mass, is given by

$$q_i^{\text{out}}(k) = \sum_{j \in \mathcal{N}_i^{\text{out}}} k_{ij} q_{i,j}(k), \tag{4}$$

where $\mathcal{N}_i^{\text{out}}$ is the set of out-neighbors of i in G_{C}. In this model, intersections are assumed to be small compared to the length of each cell, so the time spent in the intersection is negligible. The role of an efficient Autonomous Intersection Management policy is important in this assumption.

We similarly define

$$q_i^{\text{in}}(k) = \sum_{h \in \mathcal{N}_i^{\text{in}}} k_{hi} q_{h,i}(k), \tag{5}$$

where $\mathcal{N}_i^{\text{in}}$ is the set of in-neighbors of i in G_{C}.

The evolution of any cell in the network is given by

$$\rho_i(k+1) = \rho_i(k) + \frac{\Delta t}{L(\ell_p + u_p)}(q_i^{\text{in}}(k) - q_i^{\text{out}}(k)), \quad i \in C. \tag{6}$$

To enable lane reversal, the control input $u \in \mathbb{Z}^{|R|}$ determines the number of lanes per road which directly affects that road's density, see Figures 3 and 4.

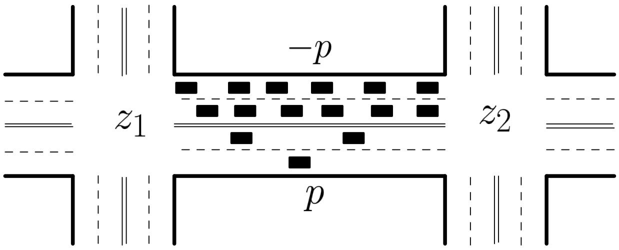

Figure 3. Before lane reversal

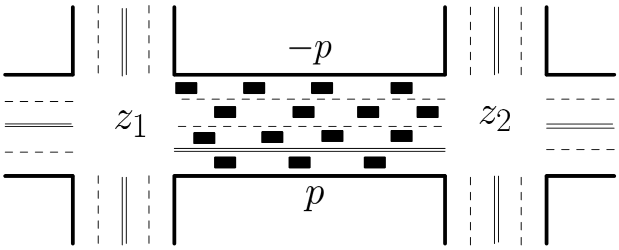

Figure 4. After lane reversal

Based on conservation of mass, cell i on side p is updated after lane reversal as follows:

$$\rho_i(k^+) = \rho_i(k) \cdot \frac{\ell_p + u_p(k)}{\ell_p + u_p(k^+)},$$

where u_p is the control before the update. For analysis purposes we assume that the change in road density is instantaneous, based on an assumption that vehicles respond quickly to a lane opening or closing. The clearing time t_c, the time it takes for all vehicles to vacate a lane being reversed, is also assumed to be zero. Lane clearing can realistically be performed in 15 seconds or less under most traffic conditions in which a lane clearing occurs, so this assumption is reasonable, Hausknecht et al. (2011a).

2.2. *Problem Formulation*

To characterize performance of the system, we define the objective function as the time spent of each vehicle in the system G_{C} summed over every vehicle, or

$$\overline{W}(u) = \sum_{w=1}^{N} \left(t_w^{\ell} - t_w^{e} \right),$$

where t_w^{ℓ} is the time in which vehicle w leaves G through a sink, t_w^{e} is the time in which vehicle w enters G through a source, and N is the total number of vehicles that spent time within the system. The total time spent is inversely proportional to the total flow rate, so total time can be approximated as

$$\begin{aligned} \overline{W}(u) &\approx \frac{N}{q_{\mathrm{avg}} \sum_{p \in S} \ell_p}, \\ &\approx \sum_{k=0}^{k_f} \left(\frac{N}{\sum_{i \in C} (q_i^{\mathrm{in}}(k) + q_i^{\mathrm{out}}(k))/2} \right), \end{aligned}$$

where q_{avg} is the average flow rate in G. The average of flow in and out of each cell is required to account for coupled dynamics.

We define the first control input as the directional lane allocation of each road $u \in \mathbb{Z}^{|R|}$, where $u_p = 1$ corresponds to reversing one lane from the default lanes in the direction of $-p$ to the direction of p in the road $r \in R$. The goal is to minimize $\overline{W}(u)$ while satisfying two physical constraints, one which maintains the total number of lanes of a roadway (the sum of lanes in both directions is constant), and the other which requires a positive integer number of lanes. This is stated as

Problem 1:

$$\begin{aligned} \underset{u \in \mathbb{R}^{|R|}}{\text{maximize}} \quad & W(u) = \sum_{k=0}^{k_f} \left(\sum_{i \in C} (q_i^{\mathrm{in}}(k) + q_i^{\mathrm{out}}(k)) \right) \\ \text{subject to} \quad & u_p \in \{-\ell_p + 1, \ldots, \ell_{-p} - 1\}, \\ & u_{-p} \in \{-\ell_{-p} + 1, \ldots, \ell_p - 1\}, \\ & u_p + u_{-p} = \ell_p + \ell_{-p}, \forall\, p, -p \in S. \end{aligned}$$

A point u^* is a critical point for Problem 1 if u^* satisfies the above constraints and if $W(u^*) \geq W(u)$ for all u s.t. $\forall\, p \in S, u$ satisfies the above constraints and $u_\zeta^* = u_\zeta, \forall\, \zeta \neq p$.

If vehicles can be redirected through intersections, then $K = \{k_{ij}\} \in \mathbb{R}^{n \times n}$ is the control variable, where k_{ij} is the proportion of vehicles flowing from cell i to cell j. Each non-zero value is lower bounded by a value $k_{\min}$ in order to maintain connectedness of the graph. Assuming a uniform critical density value in the network and given boundary conditions, maximum flow is obtained by distributing as much flow as possible into uncongested lanes. This can be formulated as

Problem 2:

$$\begin{aligned}
\underset{K\in\mathbb{R}^{n\times n}}{\text{minimize}} \quad & \tilde{W}(K) = \sum_{i\in C} \max\{0, \rho_i(t+1) - \rho_c\} \\
\text{subject to} \quad & (K\mathbf{1}_n)_i = 1, \forall\, i \notin \underline{B}, \\
& (K\mathbf{1}_n)_i = 0, \forall\, i \in \underline{B}, \\
& K \in \text{sparse}(G_{\mathrm{C}}), \\
& k_{ij} \in [k_{\min}, 1], \forall\, (i,j) \in E.
\end{aligned}$$

2.3. *Approximation of the State*

Here, we will use the assumptions employed in Hausknecht et al. (2011a) for an intersection manager (IM) to approximate the state of the traffic on the roads at the intersection. Vehicles have unique identifiers and transmit a message within $D \approx 300$ meters to the IM for a reservation request to cross intersections more efficiently. The IM at intersection $z \in Z$ maintains a counter variable $\overline{z}_p$ for road side p, adding one to $\overline{z}_p$ when it receives a notification message from a vehicle on road side p and subtracting one from $\overline{z}_p$ whenever a vehicle from road side p with a confirmed reservation is expected to leave the road and enter the intersection. The state of road side p at time t is calculated as

$$\rho_p(k) = \frac{\overline{z}_p(k)}{(\ell_p + u_p)\min\{L, D\}}. \tag{7}$$

If $L > D$ then assume that the state of the entire road is equal to the state in the nearest section. Note, with more sensing than just at intersections, the state of the roads can be more accurately approximated, so smaller cells can be used. This approximation is used in both algorithms to determine whether or not travel efficiency can be improved.

3. Lane Reversal Policy

In this section we provide a distributed Lane Reversal Algorithm together with its stability properties. The performance of the algorithm is also analyzed in Section 5.

3.1. *Lane Reversal Algorithm*

Problem 1 is a non-convex, non-smooth optimizal control problem with integer constraints. We assume that there is an intersection manager z_1 and z_2 at both ends of each road, and that z_1 is assigned its control. This IM requires estimates of the road states from its neighboring IMs and from the neighbors of z_2 to construct the complete local state. These estimates are calculated by counting vehicles in and out of each road as explained in Section 2.3 and in Equation (7).

We define $C_p = \{a \in C \mid a \text{ is a cell of } p\}$ for $p \in S$, and similarly $C_r = C_p \cup C_{-p}$, where $p, -p$ are the sides of road $r \in R$. Suppose that $T(t') \in \{1, \dots, \overline{T}\}$ represents a clock ticking from 1 to $\overline{T}$ at each IM synchronously, where $\Delta t' << \Delta t$ is a smaller discrete time step. Each IM updates its assigned roads on specific ticks, which are given by a schedule $\Lambda(r) \in \{1, \dots, \overline{T}\}$, computed during an initialization phase. As an example, in the road network in Figure 5 each road with the same number Λ can update simultaneously. By means of the flag function "`to_update`," computations are reduced to cases when changes in the neighboring conditions can lead to non-trivial updates. When the turn of an IM to update takes place (line 5), then, in order to find the best control policy for some road r with sides $p, -p \in S$ while keeping other roads fixed, $W_r(u + \omega_r \Delta_r)$ is maximized over $\omega_r \in \Omega_r$ in the Lane Reversal Algorithm. Here,

$$W_r = \sum_{k}^{k+\eta} \sum_{a \in C_{r \cup \mathcal{N}_r}} (q_a^{\text{in}}(k) + q_a^{\text{out}}(k)),$$

where η is the width of the optimization window, note that for large η, accurate prediction of the local state could require more states than just immediate neighbors. In addition, $\Omega_r = \{-\ell_p + 1, \dots, \ell_{-p} - 1\}^\eta$ is any sequence of controls and $\Delta_r \in \mathbb{Z}^n$ has zeros everywhere except 1 for each component $i \in C_p$ and -1 for each component $j \in C_{-p}$. Since in real road networks most roads have 4 or less lanes, an exhaustive search is computationally inexpensive in this domain. If a trivial update takes place, then a new update for neighboring roads is not necessary. This is encoded by setting the `to_update` function equal to zero, otherwise this function is set equal to one. State estimates are updated and information on updated controls, states and the `to_update` function is communicated to neighbors. The algorithm runs until time k_f.

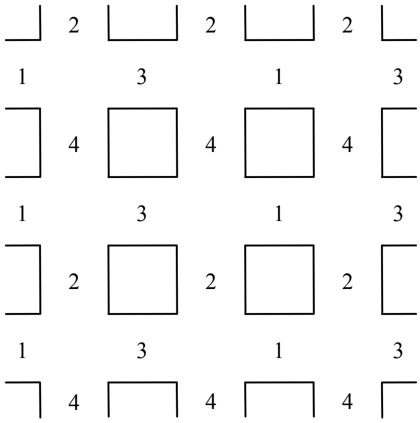

Figure 5. Schedule of road network

3.2. *Stability Analysis of Lane Reversal*

We first establish an upper bound on the objective function:

Algorithm 1: Lane Reversal Algorithm of IM z

Initialize time $t' = 0$, schedule $\Lambda(r)$, $\forall\, r \in R$;
Initialize $\texttt{to\_update}(r) = 1$, $\forall\, r \in R$;
for *all* $r \in R$ *with sides* $p, -p$ *controlled by* z *and* t' **do**
  Update $u_{p'}$, $u_{-p'}$, $\texttt{to\_update}(r')$, and $\rho_{i'}$ if messages were received from neighbors;
  if *to_update* $= 1$ *and* $\Lambda(r) = T(t')$ **then**
    $\omega_r^* \leftarrow \operatorname{argmin}_{\omega_r \in \Omega_r} W_r(u + \omega_r \Delta_r)$;
    $u_\nu^+ \leftarrow u_\nu$, $\forall\, \nu \in R \setminus \{p, -p\}$;
    $u_p^+ \leftarrow u_p + \omega_r^*$;
    $u_{-p}^+ \leftarrow u_p - \omega_r^*$;
    if $u_p^+ = u_p$ **then**
      $\texttt{to\_update}(r) \leftarrow 0$;
    else
      $\texttt{to\_update}(\lambda) \leftarrow 1$, $\forall\, \lambda \in \mathcal{N}_r$;
      $\texttt{to\_update}(r) \leftarrow 0$;
    end
    Initiate lane swap, set $\rho_i^+ = \rho_i \cdot \frac{\ell_r + u_r}{\ell_r + u_r^+}$, $\forall\, i \in C_p$ and $\rho_j^+ = \rho_j \cdot \frac{\ell_r - u_r^+}{\ell_r - u_r}$, $\forall\, j \in C_{-p}$;
    Transmit u_p^+, u_{-p}^+, $\texttt{to\_update}(\lambda) \forall\, \lambda \in \mathcal{N}_r$, $\rho_i^+ \,\forall\, i \in C_p$, and $\rho_i^+ \,\forall\, i \in C_{-p}$ values to neighbors of r;
  end
  $t' \leftarrow t' + \Delta t'$;
end

Lemma 3.1. *Under the constraints given in Problem 1, the objective function satisfies*

$$W(u) \leq 2 v_f \rho_c \ell n,$$

at each time step, assuming that $\ell_p = \ell$, $\forall\, p \in S$. This upper bound is achieved when roads are lane-balanced (for any path on the network, the number of lanes remains constant) and $\rho_i = \rho_c, \forall\, i \in C$. Intuitively, this is the state when flow through each lane is maximized over the whole network, and there is no congestion formed through lane merging. □

The proof for this lemma is omitted, as it is simply calculated by maximizing each cell's flow.

Lemma 3.2. *The* Lane Reversal Algorithm *converges in finite time to a critical point u^* of Problem 1 under the listed constraints.* □

Proof. The update u^+ is implemented in the Lane Reversal Algorithm by evaluating $W_r(u)$ and choosing ω_r which maximizes this value. Note that the algorithm constrains u_p^+ s.t. $u_p^+ \in \{1 - \ell_p, \ldots, \ell_{-p} - 1\}$. Since a local maximizer of $W_r(u)$ also maximizes $W(u)$ and the algorithm maintains a schedule which is compatible with the separability of W (no two road neighbors update simultaneously), it is guaranteed that $W(u^+) \geq W(u)$. In this way, W is a monotonically non-decreasing function through the algorithm. Using a discrete-time Lyapunov stability argument with W, asymptotic convergence to a point u^* satisfying the constraints of Problem 1 for which W can not be improved by modifying u^* entry-wise can be guaranteed. Due to the finite discrete state space, convergence occurs in finite time.

$\square$

4. Vehicle Rerouting Policy

In this section, we provide a stability analysis for the road density evolution under the dynamics (3) to (6), under the assumption of balanced sources and sinks. This motivates the distributed rerouting algorithm, which is simulated in Section 5. For this section only, we assume that there is no lane reversal occuring, and that the number of lanes on each road are equal, so $\ell_p + u_p = \ell_{p'} + u_{p'} = \ell$, $\forall\, p, p' \in S$.

4.1. *Stability Analysis of Weight-Balanced Traffic Networks*

Because one solution to Problem 2 is achieved when as many vehicles are in free flow as possible, we are interested in finding conditions that guarantee that system achieves this state naturally. Such fundamental analysis helps gain intuition and a better understanding of how the system behaves. We find one such sufficient condition here.

We define $\alpha = [\alpha_1, \dots, \alpha_n]^\top$, where $\alpha_i \in [0, 1]$ is the strength of the source connected to cell i ($\alpha_i = 0$ if no source is connected to cell i). Similarly, $\beta = [\beta_1, \dots, \beta_n]^\top$ where $\beta_i \in [0, 1]$ is the strength of the sink connected to cell i ($\beta_i = 0$ if no sink is connected to cell i). We use two assumptions:

Assumption 4.1 (Critical Density Value)**.** *The critical density value is* $\rho_c \in (0, \frac{\rho_{\text{jam}}}{2})$.

Assumption 4.2 (Weight Balanced)**.** *The graph* G_{C} *is weight balanced including boundary conditions, or equivalently,* $K^\top \mathbf{1}_n + \alpha = \mathbf{1}_n$ *and* $K\mathbf{1}_n + \beta = \mathbf{1}_n$.

Assumption 4.3 (Existence of Sources and Sinks)**.** *There exists a source and sink with nonzero strength somewhere in* G_{C}.

Note that while these are strong assumptions, the following analysis helps clarify the overall behavior of the system dynamics. Each assumption is required in the proof of Theorem 4.4. Assumption 4.1 bounds the critical density value to be between reasonable values. Assumption 4.2 requires that sinks and sources have specific coefficients to enable free flow if possible. Assumption 4.3 allows for flow into and out of the system. Under these conditions, the following result holds.

Theorem 4.4 (Stability to critical density values)**.** *Given Assumptions 4.1, 4.2, 4.3, the dynamics of Equations* (3) *through* (6) *and a connected graph* G_{C}*, any initial state with* $\rho_i(0) \in [0, \rho_{\text{jam}}], \forall\, i \in C$*, converges practically to* ρ_c*. In other words,* $\lim_{k\to\infty} \rho(k) \in [(\rho_c - \epsilon)\mathbf{1}_n, (\rho_c + \epsilon)\mathbf{1}_n]$*, where* $\epsilon \leq \frac{\Delta t v_f \rho_c}{L}$*.* $\square$

Intuitively, this result holds due to the natural dynamics as well as the existence of ideal boundary conditions.

Proof. First, we check that $\rho^* = \rho_c \mathbf{1}_n$ is indeed an equilibrium. Combining Equations (3) to (6) gives

$$\rho_i(k+1) = \rho_i(k) + \frac{\Delta t}{L\ell}\left(\sum_{a\in\mathcal{N}_i^{\text{in}}} k_{ai}\min\{\mathbf{q}_a(k), \mathbf{q}_i(k)\} - \sum_{b\in\mathcal{N}_i^{\text{out}}} k_{ib}\min\{\mathbf{q}_i(k), \mathbf{q}_b(k)\}\right).$$

The first minimum term simplifies to $v_f \ell \rho_c$ if a is not a source, and $\alpha_a v_f \ell \rho_c$ if a is a source. Similarly, the second minimum term simplifies to $v_f \ell \rho_c$ if b is not a sink, and $\beta_b v_f \ell \rho_c$ if b is a sink. Setting $\rho_i(t+1) = \rho_i(t)$ to define an equilibrium results in $K^\top \mathbf{1}_n + \alpha = K\mathbf{1}_n + \beta$. This holds given Assumption 4.2, so $\rho = \rho_c \mathbf{1}_n$ is an equilibrium.

To prove convergence to this equilibrium, we define a Lyapunov fuction

$$V(\rho) = \min_{i\in C} \begin{cases} v_f(\rho_i - \rho_c), & \rho_i \le \rho_c, \\ \frac{v_f \rho_c}{\rho_{\text{jam}} - \rho_c}(\rho_c - \rho_i), & \rho_i > \rho_c. \end{cases}$$

This function is inversely proportional to the minimum flow rate in the system, is minimized at $\rho = \rho_c$, and is non-increasing along the dynamics assuming a small Δt, we will see this by bounding the flow in and flow out of each cell. For some state ρ, define $\underline{d} = V(\rho)/v_f$ and $\overline{d} = V(\rho)(\rho_{\text{jam}} - \rho_c)/(v_f \rho_c)$. Assume for now that $\rho_c \le \rho_i \le \rho_c + \overline{d}$ for some cell i. Then based on Equations (3) to (6),

$$\begin{aligned} v_f \ell(\rho_c - \underline{d}) \le & q_i^{\text{in}}(k) \le \frac{v_f \rho_c}{\rho_{\text{jam}} - \rho_c}\ell(\rho_{\text{jam}} - \rho_i(k)), \\ v_f \ell(\overline{d} - \rho_c) \le & q_i^{\text{out}}(k) \le v_f \ell \rho_c. \end{aligned}$$

This holds regardless of the number of in-neighbors, out-neighbors, sources, and sinks that are connected to i because of Assumption 4.2. Using these inequalities, we can bound the density update as follows:

$$\rho_i(k+1) \le \rho_i(k) + \frac{v_f \Delta t}{L}\left(\frac{\rho_c}{\rho_{\text{jam}} - \rho_c}(\rho_{\text{jam}} - \rho_i(k)) - (\rho_c - \overline{d})\right).$$

Under Assumption 4.1, $\frac{\rho_c}{\rho_{\text{jam}} - \rho_c} \le 1$ which means $\rho_i(k+1) \le \rho_c + \overline{d}$ if

$$\Delta t \le \frac{L}{v_f}. \tag{8}$$

Similar arguments show $\rho_i(t+1) \ge \rho_c - \underline{d}$ when Equation (8) holds, so $V(\rho)$ is non-increasing. If initially $\rho_c - \underline{d} \le \rho_i(t) \le \rho_c$, analogous arguments using differing bounds on q_i^{in} and q_i^{out} lead to the same conclusion of Equation (8).

To prove a guaranteed decrease in $V(\rho(k))$, we must look the cardinality of $\Xi(k) = \{i \in C \mid \rho_i(k) = \max_{i \in C} \rho_i(k)\}$ and $\Phi(k) = \{i \in C \mid \rho_i(k) = \min_{i \in C} \rho_i(k)\}$, and prove that each cardinality will decrease.

Suppose there exists some $i \in \Xi(t) \cap \Xi(k+1)$ s.t. $\rho_i(k+1) = \rho_i(k) = \rho_c + \overline{d}$ with one out-neighbor j. This only occurs when $q_i^{\text{in}}(k) = q_i^{\text{out}}(k)$, and applying Equations (3) to (6), this requires $\rho_j(k) = \rho_c + \overline{d}$ where j is the out-neighbor of i. If j satisfies $\rho_j(k+1) = \rho_j(k) = \rho_c + d$ then we reapply this argument until we can find the front of group (defined when an out neighbor is not $\rho_c + \overline{d}$, or is a sink). For this cell h, we know $q_h^{\text{in}}(k) > q_h^{\text{out}}(k)$, so $\rho_h(k+1) < \rho_h(k)$ and Ξ loses a member at time $t+1$. It is possible for a node i s.t. $\rho_i(k) < \rho_c + \overline{d}$ to satisfy $\rho_i(k+1) = \rho_c + \overline{d}$, but this can only occur if each out-neighbor of i is at $\rho_c + \overline{d}$, this can be seen from Equation (4). So for each cluster of nodes in $\Xi(k)$, one (or more, at an intersection) node can join $\Xi(k+1)$, but each node that joins requires all of its out-neighbors to be at $\rho_c + \overline{d}$. On a single road side, one cell can join $\Xi(k+1)$ and one cell must leave $\Xi(k+1)$, at a normal intersection four cells can join $\Xi(k+1)$ and four cells must leave $\Xi(k+1)$, this can be also derived from Equations (3) to (6). This is a traffic congestion wave. Under Assumption 4.3, the wave will eventually propagate to a source, and once the wave is adjacent to a source, the it must be $|\Xi(k+1)| < |\Xi(k)|$ because any cell leading into the wave will not achieve $\rho_c + d$. Analogous arguments hold for Φ. Once both $\Xi(k+1)$ and $\Phi(k+1)$ are empty, then $V(\rho(k+1)) < V(\rho(k))$. This holds for sufficiently large $V(\rho(k))$, so in summary $V(\rho(k))$ approaches a neighborhood around $v_f \rho_c$ asymptotically. □

We can make a similar Lyapunov argument proving that

$$\lim_{t \to \infty} \rho_i(t) \in [\rho_{\text{src}}^{\min}, \rho_{\text{snk}}^{\max}], \quad \forall\, i \in C,$$

where $\rho_{\text{src}}^{\min} = \rho_c \min_{i \in C}(\alpha_i + \sum_{j \in C} k_{ji})$ and $\rho_{\text{snk}}^{\max} = \rho_{\text{jam}} - \rho_c \min_{i \in C}(\beta_i + \sum_{j \in C} k_{ij})$. Bounding the flow rate in and out excludes any state from being at equilibrium outside these bounds, given the previous assumptions.

Remark 4.5. Given additive zero-mean i.i.d. noise on the dynamics equation (6), it is easy to see that $V(\mathbb{E}[\rho(k+1)]) \leq V(\rho(k))$. The effect of this noise on convergence to equilibrium is tested in simulation, see Figure 8. □

Remark 4.6. The previous analysis helps motivate the rerouting algorithm. Maintaining the sufficient condition Assumption 4.2 is crucial for equilibrium behavior of the network, and it is expected that, when boundary conditions oscillate about this condition, convergence to a close-to-equilibrium condition will occur. However, within these constraints we would like to hasten convergence to further reduce delays, the benefits of which are heightened under time-varying boundary conditions.

4.2. *Rerouting Algorithm*

In some situations, it is feasible to direct vehicles where to go, in order to maintain a balanced network over time. One example is a near future road setting where driverless vehicles can be dynamically reassigned to pick up waiting passengers. Freight-type of vehicles or autonomous cars in mobility-on-demand systems could also be influenced in real traffic based on AIM-vehicle communication mechanisms. Control over vehicle direction means that K can be altered under some constraints to improve the total flow over the time interval. Because maximum flow is achieved when $\rho_i = \rho_c$, $\forall\, i \in C$, flow through intersections can be redirected from more dense roads to less dense roads, keeping the system close to ρ_c. This problem is formulated in Problem 2.

A greedy approach suits this problem because it increases immediate flow through the intersection and speeds up the balancing of neighboring roads while maintaining the equilibrium of the natural dynamics. A potential strategy is to redirect flow from all in-neighbors of an intersection to the least dense out-neighbor. If the least dense out-neighbor is more dense than ρ_c, then redirect towards a sink if possible. However, this approach creates congestion when sinks are not ideal by attempting to direct flow out of the system but the sink restricting the flow out, so we require an alternative policy. Instead, each out-neighbor is sorted according to how much flow they will allow in, and they are paired with in-neighbors which provide the most flow, see Rerouting Algorithm.

Algorithm 2: Rerouting Algorithm

for *each intersection* $z \in Z$ *at each time* k **do**
  $D^+ \leftarrow$ vector of cells flowing into z, sorted by decreasing density;
  $D^- \leftarrow$ vector of cells receiving flow from z, sorted by increasing density;
  for $\gamma \in \{1, \dots, |D^+|\}$ **do**
    $x \leftarrow D^+_\gamma$;
    $y \leftarrow D^-_\gamma$;
    for $\zeta \in \{1, \dots, n\}$ **do**
      if $k_{x,\zeta} > 0$ *and* $\zeta \neq y$ **then**
        $k_{x,\zeta} \leftarrow k_{\min}$;
      end
      if $k_{\zeta,y} > 0$ *and* $\zeta \neq x$ **then**
        $k_{\zeta,y} \leftarrow k_{\min}$;
      end
    end
    $k_{x,y} \leftarrow 1 - (|D^+| - 1)k_{\min}$;
  end
end

This algorithm is decentralized, the only information required is from immediate neighbors of an IM. Intuitively, the algorithm is improving flow by directing flow from the most dense in-neighbors of the intersection to the least dense out-neighbors so that the flow from/to the most/least congested road is relatively unrestricted. This pushes both states more quickly towards ρ_c, and because they are the furthest away from ρ_c, this helps decrease $V(\rho(k))$ more rapidly. Under varying boundary conditions, this improvement is enhanced.

5. Simulation Results

In simulation, both lane reversal and rerouting vehicles reduce overall traffic delay under imbalanced conditions as we discuss next. We use $\Delta t = 1$ second, $L = 500$

meters, $\ell = 4$ lanes per road, and $v_f = 60$ km/hr, which also satisfy Equation (8). Note, $\rho_{\text{jam}} \approx 226 \frac{\text{veh}}{\text{km}\cdot\text{lane}}$ was calculated by assuming an average vehicle length of 4.11 meters and an average gap between stationary vehicles of 0.31 meters. We chose $\rho_c = \rho_{\text{jam}}/3$.

Lane Reversal Algorithm

Generally, lane reversal creates a significant short-term improvement of traffic flow. Lane reversal can hinder overall traffic flow in the long-term when reversing a lane requires lane merging somewhere else in the system or when the road side with less lanes receives heavy traffic flow afterwards. This first issue can be addressed through coupling roads together so that their control variables are equal and no lane merging is required between them, in simulation we choose the more conservative control value from both intersections and apply that to both roads. The second issue can be at least partially addressed through predicting traffic patterns using past data or communications with a larger portion of the road network, though we do not address this in this paper.

We simulated a two road network with an intersection between them, sources and sinks at the boundary, and an initial state which had light congestion randomly sampled from $[0, \rho_{\text{jam}}/2]$ on one side of both roads and heavy congestion randomly sampled from $[\rho_{\text{jam}}/2, \rho_{\text{jam}}]$ on the other. Instead of optimizing over a time horizon, a greedy algorithm was enough to see significant improvements. The boundary conditions α and β on one end of the network were randomly sampled from $[0, 1]$ and $\alpha = 1$ and $\beta = 1$ on the other side, creating an imbalanced flow. U-turns do not occur, and flow from each road to any neighbor is equally likely. One example of the benefit of lane reversal on the objective function is shown in Figure 6, it is evident that the flow rate increases immediately after lane reversal. The throughput improvement of 100 experiments is shown in Figure 7, we can see a large variance in the data but there is always significant improvement. Note, maintaining a constant number of lanes along all paths becomes impossible with a larger system, creating merges which can negatively affect the overall equilibrium.

We also added zero-mean Gaussian noise to Equation (6) to check the robustness of Theorem 4.4 (Stability to critical density values), the effect of noise on the two road network is seen in Figure 8. Deviation from equilibrium was averaged over each cell and over 120 time steps, with the initial state at equilibrium. With zero noise there is no deviation from equilibrium, and as the noise level increases, the average deviation from equilibrium increase quite linearly from small noise values.

Rerouting Algorithm

The rerouting policy improves flow rate in both short term and long term while maintaining the original equilibrium, though in a less dramatic fashion than lane reversal. We have simulated a two block by two block road network with random initial densities under random constant boundary conditions with $k_{\min} = 0.05$. We implemented the rerouting policy on the central intersection, see Figure 9, black represents very dense and white represents no vehicles. This shows an initially unbalanced state which converged to a more efficient state with help from the rerouting policy in the middle

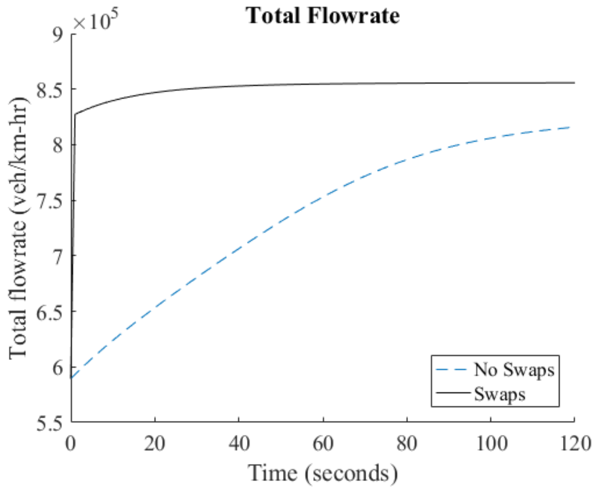

Figure 6. Lane reversal on two roads. The solid line represents lane reversal and the dashed represents no lane reversal, note the immediate improvement. In this example, one lane was reallocated from westbound to eastbound at $t = 0$ then the state remained constant.

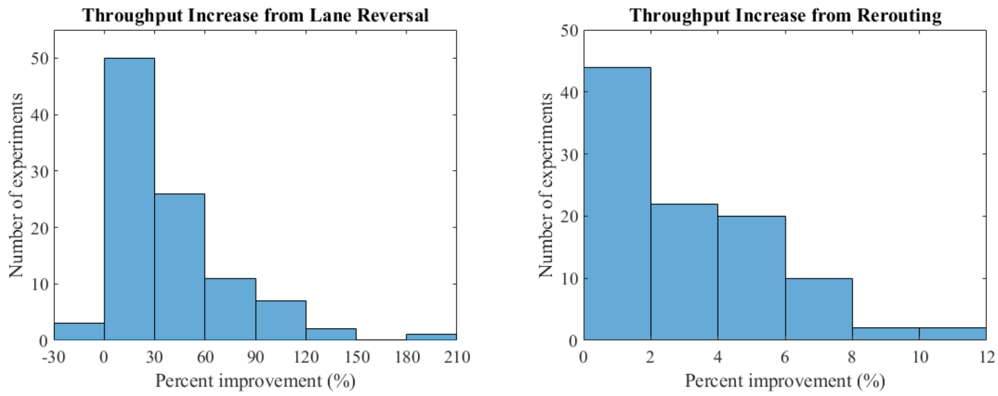

Figure 7. Relative improvement under lane reversal and rerouting algorithm, respectively.

intersection, see 10.

In Figure 7, we can see relative improvements on the total flow rate given by this policy, ranging from negligible to moderate. We note that the objective function describes the performance of the each cell in the system, and in this network there are 240 cells, only 8 of which are connected to the intersection performing the policy, so only moderate improvement is expected. Under specific initial conditions, the most improvement was seen at 36%, on average improvement is between 0 and 8%. In every case, this algorithm improved overall flow rate.

6. Conclusion and Future Work

In conclusion, we extended the cell transmission model and established objective functions with the goal of minimizing total time spent on the road. We proposed a distributed algorithm to efficiently calculate and implement an appropriate lane direction

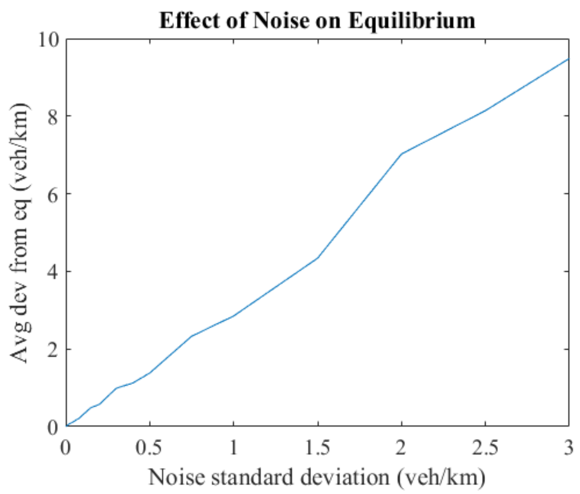

Figure 8. Here, the effect of noise on the equilibrium at critical density is shown.

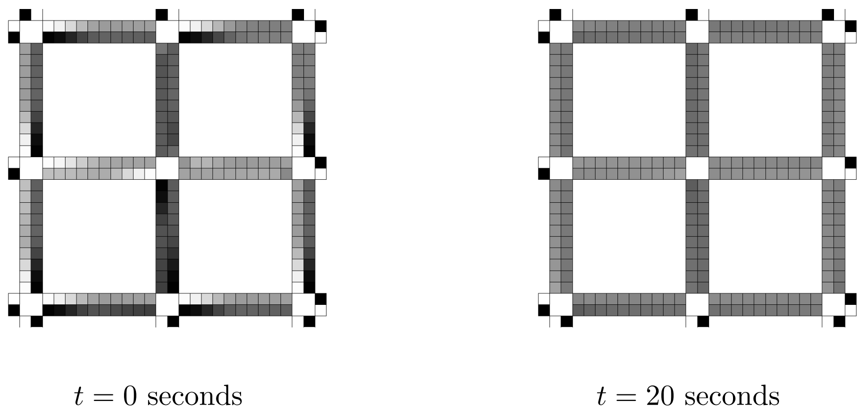

Figure 9. Twelve Road Network

reallocation. We also proposed a distributed algorithm to dynamically reroute vehicles to improve the long term behavior of the system. We proved convergence of the lane reversal algorithm to a critical point and bound the equilibria of the traffic rerouting algorithm under certain conditions. We showed through simulations performance gains using lane reversal on a network under particular conditions and using rerouting under different conditions.

There are many avenues for future work on this problem. One avenue is improving the traffic model and comparing its predictions with real traffic traffic data to ensure accuracy, in particular the vehicle time spent in an intersection is currently assumed constant. A microscopic model will better capture the dynamics of real vehicles on a road network, for example by implementing simulations which include spawning and tracking individual vehicles. Measurement and model uncertainty could be characterized for more accurate estimation. Reinforcement learning can address some of the issues with using a greedy lane reversal algorithm to further reduce total time delays.

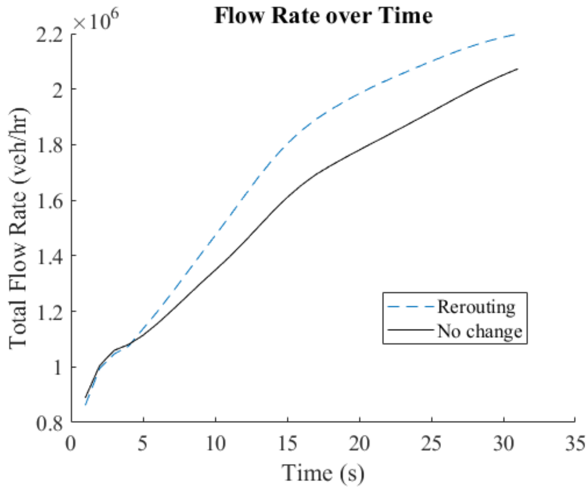

Figure 10. Rerouting on grid network. The dashed line represents rerouting and the solid line represents no rerouting, note the consistent improvement.

References

A. Abadi, P. A. Ioannou, and M. M. Dessouky. Multimodal dynamic freight load balancing. *IEEE Transactions on Intelligent Transportation Systems*, 2015. Scheduled for publication.

T. C. Au, N. Shahidi, and P. Stone. Enforcing liveness in autonomous traffic management. In *AAAI Conference on Artificial Intelligence*, pages 1317–1322, 2011.

G. Bretti, R. Natalini, and B. Piccoli. Numerical approximations of a traffic flow model on networks. *Networks and Heterogeneous Media*, 1(1):57–84, 2006.

M. Campbell, M. Egersdedt, J. P. How, and R. M. Murray. Autonomous driving in urban environments: approaches, lessons and challenges. *Philosophical Transactions of the Royal Society A*, 368(1928):4649–4672, October 2010.

D. Carlino, S. D. Boyles, and P. Stone. Auction-based autonomous intersection management. In *IEEE Int. Conf. on Intelligent Transportation Systems*, pages 529–534, 2013.

Y. C. Chiu, H. Zheng, J. A. Villalobos, W. Peacock, and R. Henk. Evaluating regional contraflow and phased evacuation strategies for texas using a large-scale dynamic traffic simulation and assignment approach. *Journal of Homeland Security and Emergency Management*, 5 (1), 2008.

C. F. Daganzo. The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation Research Part B: Methodological*, 28(4):269–287, 1994.

D. Downey, 2008. http://www.sandiegouniontribune.com/sdut-region-caltrans-rolls-out-barrier-moving-machine-2008aug18-story.html.

K. Dresner and P. Stone. A multiagent approach to autonomous intersection management. *Journal of Artificial Intelligence Research*, 31:591–656, 2008.

E. Gravelle and S. Martínez. Traffic delay reduction via dynamic lane reversal and rerouting distributed algorithms. In *Mathematical Theory of Networks and Systems*, Minneapolis, MN, USA, July 2016.

M. Hausknecht, T. C. Au, and P. Stone. Autonomous intersection management: Multi-intersection optimization. In *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, pages 4581–4586, 2011a.

M. Hausknecht, T. C. Au, and P. Stone. Dynamic lane reversal in traffic management. In *IEEE Int. Conf. on Intelligent Transportation Systems*, pages 1929–1934, 2011b.

M.J. Lighthill and G.B. Whitham. On kinematic waves. II. a theory of traffic flow on long crowded roads. *Royal Society of London. Proceedings Series A: Mathematical, Physical and*

Engineering Sciences, 229(1178):317–345, 1955.

Q. Meng, H. Khoo, and R. Cheu. Research on reversal lane application method of urban road network based on the bi-level programming. *Advances in Intelligent Systems and Computing*, 279:983–992, 2014.

P. I. Richards. Shock waves on the highway. *Operations Research*, 4(1):42–51, 1956.

P. Tallapragada and J. Cortés. Coordinated intersection traffic management. *IFAC-PapersOnLine*, 48(22):233–239, 2015. *IFAC Workshop on Distributed Estimation and Control in Networked Systems*, Philadelphia, PA.

J. W. Wang, H. F. Wang, W. J. Zhang, W. H. Ip, and K. Furata. Evacuation planning based on the contraflow technique with consideration of evacuation priorities and traffic setup time. *IEEE Transactions on Intelligent Transportation Systems*, 14(1):480–485, 2012.

J. T. Wen and M. Arcak. A unifying passivity framework for network flow control. *IEEE Transactions on Automatic Control*, 49(2):162–174, 2004.

D. B. Work, S. Blandin, O. P. Tossavainen, and A. M. Bayen. A distributed highway velocity model for traffic state reconstruction. *Applied Mathematics Research eXpress (AMRX)*, pages 1–35, 2010.

C. Wuthishuwong and A. Traechtler. Coordination of multiple autonomous intersections by using local neighborhood information. In *Int. Conf. on Connected Vehicles and Expo*, pages 48–53, 2013.

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*, 8(1):183–191, 2000.

W. W. Zhou, P. Livolsi, E. Miska, H. Zhang, J. Wu, and D. Yang. An intelligent traffic responsive contraflow lane control system. In *Vehicle Navigation and Information Systems Conference*, pages 174–181, 1993.