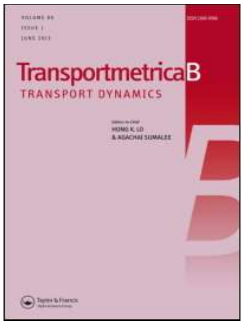

Taylor & Francis
Taylor & Francis Group

Transportmetrica B: Transport Dynamics

ISSN: 2168-0566 (Print) 2168-0582 (Online) Journal homepage: https://www.tandfonline.com/loi/ttrb20

Max-pressure control of dynamic lane reversal and autonomous intersection management

Michael W. Levin, David Rey & Adam Schwartz

TRANSPORTMETRICA B: TRANSPORT DYNAMICS
2019, VOL. 7, NO. 1, 1693–1718
https://doi.org/10.1080/21680566.2019.1691675

Max-pressure control of dynamic lane reversal and autonomous intersection management

Michael W. Levin[a], David Rey[b] and Adam Schwartz[c]

[a]Department of Civil, Environmental, and Geo- Engineering, University of Minnesota, Minneapolis, MN, USA;
[b]Research Centre for Integrated Transport Innovation (rCITI), University of New South Wales, Sydney, Australia;
[c]School of Public Health, University of Minnesota, Minneapolis, MN, USA

ABSTRACT

Autonomous intersection management (AIM) (which coordinates intersection movements to avoid signal phases) and dynamic lane reversal (DLR) (which frequently changes lane directions in response to time-varying demand) have previously been proposed for connected autonomous vehicles. A major open question for both is finding the optimal control policy. This paper develops a decentralized max-pressure policy that controls both AIM and DLR based on queue lengths on adjacent links. Using a stochastic queueing model, we prove that the max-pressure policy is also *throughput-optimal*; any demand that can be stabilized (queue lengths remain bounded) will be stabilized by the max-pressure policy. We show numerically that DLR significantly increases the stability region, particularly for asymmetric demand. Since the stochastic queueing model excludes some realistic aspects of traffic flow, we adapt the max-pressure control for simulation-based dynamic traffic assignment. Results on a city network show significant improvements from max-pressure AIM with and without DLR.

ARTICLE HISTORY
Received 27 April 2018
Accepted 6 November 2019

KEYWORDS
Autonomous intersection management; dynamic lane reversal; dynamic traffic assignment; cell transmission model; max-pressure control

1. Introduction

Revolutionary new traffic control technologies have been proposed for autonomous vehicles (AVs) to increase traffic flow and reduce congestion. The two such technologies that we address are autonomous intersection management (AIM) and dynamic lane reversal (DLR). AIM (Dresner and Stone 2004, 2006) is an alternative to traffic signals which coordinates the movement of individual AVs. AVs communicate their requested turning movement to the *intersection manager*, which microsimulates the requests of all vehicles waiting at each intersection to find non-conflicting combinations of turning movements. The solutions are then communicated back to each AV in the form of intersection access time and speed. By controlling individual vehicles, AIM greatly expands the feasible region for intersection movements, and previous work has found that simple policies for AIM (such as first-come-first-served) could perform better than optimized traffic signals for some intersections (Fajardo et al. 2011; Li et al. 2013).

DLR is an extension of contraflow lanes for AVs. Contraflow lanes are typically applied on a peak hour schedule (e.g. Zhou et al. 1993; Xue and Dong 2000; Meng and Ling Khoo 2008) or for evacuations (e.g. Zhang and Bing Lei Xie 2012; Wang et al. 2013; Dixit and Wolshon 2014) because lane reversals can cause major safety problems for human drivers. However, lane direction changes could be communicated directly to individual AVs through infrastructure-to-vehicle communications to

CONTACT Michael W. Levin ✉ mlevin@umn.edu

admit more frequent spatio-temporal changes in lane direction. Hausknecht et al. (2011) proposed to use infrastructure for AIM to enable DLR. Since DLR requires more complex communications and coordination than AIM, models of DLR have typically used AIM for the intersection control (Hausknecht et al. 2011; Levin and Boyles 2016).

Both AIM and DLR expand the feasible region for traffic control, and are sufficiently novel that the question of control policy has been studied but is still open. For instance, initial work on AIM used the first-come-first-served policy (Dresner and Stone 2004; Fajardo et al. 2011) with the goal of minimizing delay. Initial work on DLR has used saturation-based heuristics for lane allocation (Hausknecht et al. 2011; Levin and Boyles 2016). For both AIM and DLR, the problem of finding the optimal control given known deterministic demand has been formulated as mixed integer linear programs (Duell et al. 2016; Levin, Fritz, and Boyles 2017). However, due to the computational requirements of solving such formulations, previous numerical results have either been limited to small networks or relied on heuristics.

The ideal solution is to find a throughput-optimal decentralized policy that does not assume to know future travel demand (which depends on vehicle departures and route choices). A *decentralized* policy can be computed at the individual intersection level without knowledge or consideration of the state elsewhere, and is therefore desirable for practical implementation. Optimality includes several possible objectives, such as maximum throughput or minimum delay. The purpose of this paper is to develop a throughput-optimal decentralized control policy for both DLR and AIM. As we will demonstrate, the AIM and DLR control problems can be formulated together in a manner highly similar to the signal timing model of Varaiya (2013), which admits a similar proof of throughput optimality. Nevertheless, the formulation and proof are valuable to answer the open questions of how to control AIM and DLR.

This paper makes two main contributions. First, we formulate the problem of controlling AIM and DLR as a stochastic queueing model, and define a max-pressure policy that is proven to be throughput-optimal. This is the first max-pressure policy for either AIM or DLR.

We then compare the stability region (the set of demands for which bounded queue lengths are possible) for AIM, DLR with AIM, and traffic signals to explore whether AIM and DLR would reduce queues in the traffic network. Second, since the queueing formulation makes unrealistic assumptions on the traffic flow behavior, we apply the max-pressure policy to simulation-based dynamic traffic assignment (DTA). Although the policy is not provably optimal in DTA, we demonstrate the effectiveness of max-pressure control on a calibrated city network.

The remainder of this paper is organized as follows: Section 2 describes the state of the art on control of AIM and DLR, and discusses previous work on max-pressure control. Section 3 formulates the stochastic queueing model and gives the max-pressure control. We investigate the stability region in Section 4. Section 5 describes the simulation-based DTA model used to study max-pressure control under more realistic traffic flow and route choice behavior. We present numerical results on a city network in Section 6, and conclude in Section 7.

2. Background

AIM has received considerable attention in the literature, including several different studies on control policies (Section 2.1). DLR has received relatively less attention, but makes use of technologies similar to those required for AIM (Section 2.2). Although pressure-based control has yet to be studied for AIM or DLR in the literature, it has been previously been applied to signalized intersection control (Section 2.3).

2.1. Autonomous intersection management

AIM was first proposed by Dresner and Stone (2004, 2006) as an alternative to traffic signals, and similar technologies have been tested in autonomous vehicles (Alonso et al. 2011). Vehicles use vehicle-to-infrastructure communications to transmit a request for a specific turning movement and entry time to the intersection manager, which uses forward simulation to accept a subset of non-conflicting requests. A key question is how to decide which request to reject when vehicle requests conflict. Initial work focused on the control protocol and used a first-come-first-served (FCFS) policy: the vehicle that sent its request first receives priority in request acceptance. In addition to being fair, the FCFS policy fits in well with the AIM protocol. Fajardo et al. (2011) and Li et al. (2013) found that the FCFS policy could perform better than traffic signals for a variety of demand scenarios. However, FCFS can easily become worse than traffic signals for asymmetric intersection geometries or by breaking intersection coordination (Levin, Boyles, and Patel 2016). Since AIM can mimic traffic signals (Dresner and Stone 2007), the feasible region of AIM includes traffic signals. Therefore, AIM can always perform at least as well as traffic signals. Previous studies have investigated alternatives to FCFS for prioritizing vehicle movement.

Schepperle and Böhm (2007) and Vasirani and Ossowski (2012) suggested holding auctions at each intersection to determine which vehicles received priority. However, it is not clear whether these auctions significantly improved over FCFS, since they essentially just create a random priority of vehicles at the intersection. Carlino, Boyles, and Stone (2013) used system bids to augment auctions and reduce travel times, but it was not clear if the auctions themselves had a significant impact on the observed improvements. Levin and Boyles (2015) found that many high-bidding vehicles became stuck behind low-bidding vehicles, reducing the benefits of auctions for high-bidders.

The more interesting question is how to find the *optimal* ordering of vehicles through an intersection. Using AIM, individual vehicles can be controlled, allowing many more combinations of vehicle ordering than with traffic signals. Tachet et al. (2016) proposed modifying a platoon-based policy for granting reservations to batches of vehicles moving together. Zhu and Ukkusuri (2015) developed a linear program for AIM within DTA, although it was limited to one vehicle per conflicting movement per time step. Levin, Fritz, and Boyles (2017) formulated the vehicle ordering problem as a mixed integer program, again for a single intersection within DTA. Since DTA operates on time steps, and is not necessarily practical for actual control, Levin and Rey (2017) developed a mixed integer linear program for optimizing vehicle trajectories in continuous-time. They also modified the original AIM protocol (Dresner and Stone 2004) to admit optimized trajectories.

A major downside of AIM is the lack of support for human-driven (legacy) vehicles. Consequently, AIM is unlikely to be usable until AVs comprise most of the vehicle traffic. Dresner and Stone (2007) proposed occasionally cycling a green traffic light to clear queues of legacy vehicles, but delays could be high. Conde Bento et al. (2013) proposed to reserve all possible vehicle trajectories for legacy vehicles, creating a method for legacy vehicles to use the reservation system without two-way communications and precise control. However, due to the extra space reserved for legacy vehicles, this method is unlikely to improve over traffic signals until reaching 80% or higher AV market penetration (Levin and Boyles 2016). Qian et al. (2014) suggested that legacy vehicles might follow AVs through an intersection. The problem of integrating legacy vehicles into AIM safely yet efficiently is challenging and requires further study. It is also tangential to the purpose of this paper, which is to optimize AIM and DLR for the best case scenario of 100% AV market penetration.

2.2. Dynamic lane reversal

AIM improves intersection capacity, but does little for link capacity. Hausknecht et al. (2011) proposed using the intersection manager technology for AIM to control lane access as well. Lane direction becomes dynamic because vehicles must request permission to use a specific lane in a specific direction as well. DLR is similar to contraflow lanes currently in use in some cities during peak hours or for evacuations. However, the main difference between DLR and contraflow lanes is that two-way communications and precise control of AVs might be used to change lane direction at much more frequent intervals, such as on the order of minutes (Hausknecht et al. 2011). Hausknecht et al. (2011) suggested changing the direction of the lane for an entire link, but lane direction could also be modified at smaller spatial intervals. For instance, Duell et al. (2016) proposed a mixed integer linear program for determining lane direction in system optimal DTA. Levin and Boyles (2016) studied DLR in (user equilibrium) DTA, and developed a saturation-based heuristic for determining lane direction based on expected future demand rates and turning proportions. However, they were not able to find an optimal policy efficiently, especially when future demand was not known with certainty. This paper improves significantly on previous methods. The max-pressure control is provably optimal for a point-queue based flow model, like the one used by Varaiya (2013), and does not require any knowledge of future demand. The policy is used to develop a heuristic for simulation-based DTA using the cell transmission model, which is shown to be effective on a city network.

2.3. Pressure-based control

Max-pressure control is inspired by the backpressure policy for internet communications (Tassiulas and Ephremides 1992). The control is responsive to the queue lengths, which are used to determine the pressure for specific turning movements. Typically, pressure for movement from queue *i* to queue *j* increases with the length of *i* and decreases with the length of *j*. The objective of pressure-based control is to maximize the stability of the network, where stability is defined in terms of the sum of the queue lengths. If queues grow to infinity, the network is said to be unstable. To apply max-pressure control, the traffic network is typically modeled as a stochastic queueing system, and stability is proven using Lyapunov analysis. Varaiya (2013) studied the variant in which a single phase is chosen per time step. Le et al. (2015) created a similar max-pressure signal control policy, instead choosing the fraction of each time step allocated to each signal phase. Le et al. (2017) extended the model to include system optimal route choice and spatially-constrained queues.

In general, the stochastic queueing models used to prove the optimality of max-pressure control make several assumptions that are not applicable to traffic networks. First, they assume that queues are unbounded, whereas roads have finite space to store vehicles. Second, they typically assume that link travel times are an uniform one time step. Although that is reasonable for internet packet routing with information traveling at the speed of light, that is not applicable to traffic networks. This assumption can be somewhat overcome by breaking each link into segments, with each one traversable in one time step at the free flow speed. However, the third limiting assumption is not modeling the congested side of the flow-density relationship. Consequently, max-pressure control has yet to be proven to be throughput-optimal for realistic traffic networks, and such a proof may not exist given the challenging analytical properties of traffic flow models such as the cell transmission model (Daganzo 1994).

Nevertheless, max-pressure control has elegant analytical properties for the simplified queueing model (Varaiya 2013). To address the issues with realistic traffic flow, this paper takes a two-phase approach: we first formulate and prove stability of max-pressure control under the necessary assumptions. Then, we apply the max-pressure control to simulation-based DTA to study whether it is effective in a more realistic flow model.

Another limiting assumption made by some previous work on max-pressure control is that route choices remain fixed. Smith (1979) demonstrated that intersection control can have significant and adverse effects on route choice. Although some papers used simulation to investigate the effects of max-pressure control with route choice (Zhang, Li, Feng, and Jiang 2012; Gregoire et al. 2014; Zaidi, Kulcsár, and Wymeersch 2016), Levin, Boyles, and Patel (2016) showed that decentralized control (which includes most max-pressure policies) could create the Daganzo (1998) paradox. Despite the lack of optimality for realistic conditions due to traffic flow modeling or route choice, max-pressure control could yet be a better policy for realistic networks than FCFS or other heuristics. Therefore, this paper will develop and prove the optimality of a max-pressure policy for a stochastic queueing model with fixed routing proportions, then apply it within simulation-based DTA to calibrated network data.

3. Max-pressure control

Although max-pressure control has been formulated and proven for traffic signals (Varaiya 2013; Le et al. 2015), it has yet to be applied to AV technologies. AVs are more amenable to max-pressure control than signals because the two-way communications and potential behavioral precision of AVs admits a greater feasible range of intersection controls. For instance, signals are limited to selecting phases, whereas AIM can grant intersection access to individual vehicles (Fajardo et al. 2011). In this section, we formulate a combined max-pressure policy for controlling both AIM and DLR. Essentially, we show that we can formulate the AIM and DLR problem so similarly to the signal control model of Varaiya (2013) that the proofs of optimality are identical.

3.1. *Traffic network*

Consider a traffic network $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ with nodes $\mathcal{N}$ and links $\mathcal{A}$. Let $\Gamma^-(n) \subset \mathcal{A}$ and $\Gamma^+(n) \subset \mathcal{A}$ be the sets of links incoming and outgoing from node n, respectively. In an intuitive overload of notation, let $\Gamma^-(i)$ and $\Gamma^+(i)$ be the sets of links incoming and outgoing from link i as well. For the purposes of proving throughput optimality, links are modeled as store-and-forward queues with changing numbers of lanes, resulting in changing capacities. A *turning movement* $(i,j) \in \Gamma^-(n) \times \Gamma^+(n)$ indicates movement from link i to link j across some node n.

Consider discretized time. Let $x_{ij}(t) \in \mathbb{R}_+$ be the number of vehicles waiting for turning movement (i,j) at time step t. Let $\mathbf{x}(t)$ be the array of queue lengths at time t, and let $\mathcal{X}$ be the state space – the set of possible queue length arrays. Queue lengths evolve over time based on the intersection control and number of lanes available for vehicle movement. At the start of each time step, two controls are chosen: lane allocations and intersection flows.

Let $\tilde{\ell}_i$ be the number of lanes on link i without lane reversals. Let $\ell_i^{\uparrow}(t) \in \{0, \dots, \tilde{\ell}_i\}$ and $\ell_i^{\downarrow}(t) \in \{0, \dots, \tilde{\ell}_i\}$ be the number of lanes allocated for inflow and outflow from link i at time t, respectively. The ↑ and ↓ refer to the upstream and downstream ends of the link, respectively. As discussed by Levin and Boyles (2016), the number of lanes need not be uniform throughout a link.

Lanes may be shared between parallel, opposite-direction pairs of links. (Note that a link could have 0 lanes allocated as long as it is empty of vehicles.) Let $\mathcal{A}_{\text{dlr}} \subset \mathcal{A}$ be the set of links which have an opposite-direction pair that admits DLR. Let $i^{\leftarrow}$ be the opposing link of i. The set $\mathcal{A}_{\text{dlr}}$ is symmetric; if $i \in \mathcal{A}_{\text{dlr}}$ then $i^{\leftarrow} \in \mathcal{A}_{\text{dlr}}$ also. Furthermore, applying the $(\cdot)^{\leftarrow}$ function twice results in the original link: $(i^{\leftarrow})^{\leftarrow} = i$. Let $\mathcal{A}_{\text{fix}} = \mathcal{A} \backslash \mathcal{A}_{\text{dlr}}$ be the set of links without a parallel pair. For all links $i \in \mathcal{A}_{\text{fix}}$, $\ell_i^{\uparrow}(t) = \ell_i^{\downarrow}(t) = L_i$. Let $\mathcal{A}_{\text{r}} \subset \mathcal{A}_{\text{fix}}$ be the set of source links on which vehicles can enter the network, and let $\mathcal{A}_{\text{s}} \subset \mathcal{A}_{\text{fix}}$ be the set of sink links on which vehicles can exit. Vehicles that enter a sink link are immediately removed from the network. The set of internal links is denoted $\mathcal{A}_{\text{o}} = \mathcal{A} \backslash (\mathcal{A}_{\text{s}} \cup \mathcal{A}_{\text{r}})$.

3.2. *Lane allocation*

We do not include physical space constraints in the max-pressure formulation so that we can prove stability. Therefore, the lane allocation has only a limited effect on the space available for a link queue. In this queuing model, vehicles can 'stack' infinitely. The spatial constraints on lane allocation are

$$\ell_i^{\uparrow}(t) \geq 1 \text{if} \sum_{j \in \mathcal{A}} x_{ij}(t-1) > 0 \tag{1a}$$

$$\ell_i^{\downarrow}(t) \geq 1 \text{if} \sum_{j \in \mathcal{A}} x_{ij}(t) > 0 \tag{1b}$$

which ensures that the link is given at least one lane if a queue exists on that link. If no queue exists, then it is possible to reverse all lanes on a link, making it impassable until the next time step.

Each link and its opposite have $\tilde{\ell}_i + \tilde{\ell}_{i^\leftarrow}$ lanes when combined that can be reversed as needed. Therefore, the lane allocation is also constrained by

$$\ell_i^\uparrow(t) + \ell_{i^\leftarrow}^\downarrow(t) = \tilde{\ell}_i + \tilde{\ell}_{i^\leftarrow} \tag{2}$$

Let $\boldsymbol{\ell}(t)$ be the array of lane allocations at time t. Let $\mathcal{L}(\mathbf{x}(t))$ be the set of lane allocations satisfying constraints (1) and (2) and the domain of $\ell_i^\uparrow(t)$ and $\ell_i^\downarrow(t)$ for a given queue array $\mathbf{x}(t)$. The feasible lane allocations depend on $\mathbf{x}(t)$ due to constraints (1). The saturation flows for turning movement (i,j) will depend on the number of lanes allocated to both i and j. For convenience, let $\ell_{ij}(t) = \min\{\ell_i^\downarrow(t), \ell_j^\uparrow(t)\}$ be the number of lanes available for movement from i to j.

If a system is not capable of dynamic lane reversal, this formulation can still be used for intersection control only (see Section 3.4) by setting $\ell_i^\uparrow(t) = \ell_i^\downarrow(t) = \tilde{\ell}_i$ for all t.

An interesting property of this formulation is that $\ell_i^\uparrow(t)$ and $\ell_i^\downarrow(t)$ must be controlled separately to achieve a policy decentralized at the node level. If $\ell_i^\uparrow(t)$ and $\ell_i^\downarrow(t)$ must be equal but can vary over time, then the number of lanes allocated to outflow from link i affects the number of lanes allocated to inflow into link i. However, inflow and outflow to i occur at different intersections, which prevents decentralization by node. Levin and Boyles (2016) proposed to enact dynamic lane reversal at small spatial intervals within each link, which would satisfy the requirement that $\ell_i^\uparrow(t)$ and $\ell_i^\downarrow(t)$ are separate.

3.3. Saturation flow

Although the lane allocation does not affect the available space, it affects the available capacity for inflow and outflow. Let Q_{ij} be the saturation flow *per lane* for moving from link i to link j at time t across an intersection independent of conflicting movements. This saturation flow is based on the link capacities, but may be further limited by the turning geometry. The actual capacity for each turning movement at time t also depends on the lane allocation. For movement from i to j, the saturation flow is equal to $Q_{ij}\ell_{ij}(t) = Q_{ij}\min\{\ell_i^\downarrow(t), \ell_j^\uparrow(t)\}$ at time t. Therefore, choosing $\ell_j^\uparrow(t) = \ell_i^\downarrow(t)$ may be necessary to maximize the utilization of turning movement (i,j). However, the choice of $\ell_j^\uparrow(t)$ is also limited by constraint (2), and affects outflow on link $j^\leftarrow$.

3.4. Intersection flow

Since DLR requires 100% autonomous vehicles, and DLR systems may make use of autonomous intersection control technologies (Hausknecht et al. 2011), we combine DLR with AIM. Early versions of AIM relied on microsimulation of vehicle movements on a spatial-temporal grid per intersection. Instead of microsimulating vehicle movements within AIM, we use the conflict region model of AIM established by Levin and Boyles (2015) to model the intersection flows per time step. Previous work on max-pressure control (e.g. Varaiya 2013) similarly used a mesoscopic flow model for traffic signal flows.

Consider a time step t, and suppose that the saturation flows Q_{ij} are realized and known. Since AIM requires two-way communication with vehicles requesting intersection access, saturation flows should be known at the beginning of each time step through the reservation process. Divide each intersection into a set of *conflict regions*. One method for creating this division is to align the conflict region boundaries with incoming and/or outgoing links (Levin and Boyles 2015). Figure 1 provides an example of conflict regions. Notice that the shape of the conflict regions changes in response to lane allocation. For instance, in Figure 1, east-west movements are allocated only one lane while west-east movements are allocated three lanes. Thus the conflict regions associated with the east-west and west-east divisions are scaled proportionally.

Figure 1 illustrates that turning movements may be compressed into a single lane, depending on the lane allocation. This may result in first-in-first-out (FIFO) restrictions on exiting flow. It is not yet known how to include these restrictions in max-pressure control. Previous work (e.g. Varaiya 2013; Le et al. 2015) assumed that each turning movement had a separate queue. Because FIFO is a real

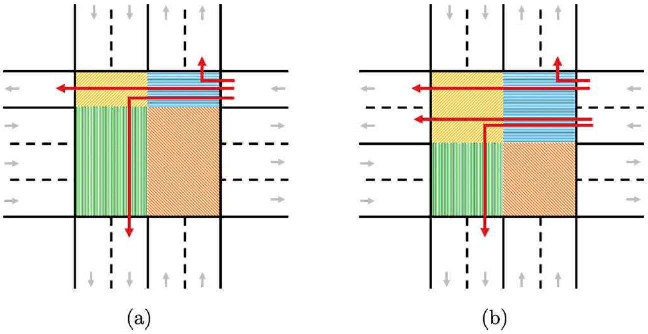

Figure 1. Illustration of conflict regions. Conflict region boundaries change in response to lane allocation, but turning movements pass through the same conflict regions regardless of the lane directions.

limitation, Section 6 applies the max-pressure control developed here to a cell transmission model-based simulation which includes FIFO.

Figure 1 also illustrates that each turning movement passes through one or more conflict regions, which restrict the total flow passing through. Let $\mathcal{R}$ be the set of all conflict regions, and let $\mathcal{R}_{ij} \subseteq \mathcal{R}$ be the subset of conflict regions used for turning movement (i,j). Assume that $\mathcal{R}_{ij}$ is independent of the lane allocation, which means that turns from i to j pass through the same conflict regions regardless of the lane allocation. That is accomplished when the conflict region geometry varies with the turn geometry, as shown in the example in Figure 1. The total flow through each conflict region is bounded, which places additional constraints on the feasible combinations of intersection flows. For instance, simultaneous east-west and west-east movements do not overlap in any conflict regions, so the conflict region constraints would be inactive. On the other hand, simultaneous west-east and north-south movements would overlap in one conflict region. Let δ^r_{ij} indicate whether turning movement (i,j) uses conflict region r:

$$\delta^r_{ij} = \begin{cases} 1 & \text{if } r \in \mathcal{R}_{ij} \\ 0 & \text{else} \end{cases} \tag{3}$$

Let $y_{ij}(t)$ be the flow from i to j. Levin, Fritz, and Boyles (2017) derived the constraint on conflict region flows as

$$\sum_{(i,j)\in\mathcal{A}^2} \delta^r_{ij} \frac{y_{ij}(t)}{\ell_{ij}(t)Q_{ij}} \leq 1 \tag{4}$$

for all $r \in \mathcal{R}$.

The objective of AIM is to decide the flows per turning movement, $y_{ij}(t)$. Finding $y_{ij}(t)$ is equivalent to finding the fraction of the time step $s_{ij}(t) \in [0,1]$ devoted to movement from i to j. $y_{ij}(t)$ relates to $s_{ij}(t)$ via

$$y_{ij}(t) \leq s_{ij}(t)Q_{ij}\ell_{ij}(t) \tag{5}$$

If $s_{ij}(t) = 1$, then flow from i to j can be equal to the saturation flow. Equations (4) and (5) can be combined to obtain

$$\sum_{(i,j)\in\mathcal{A}^2} \delta^r_{ij} s_{ij}(t) \leq 1 \tag{6}$$

We formulate a mathematical program to find the best $s_{ij}(t)$ for a single time step t given some weight per turning movement $w_{ij}(t)$:

$$\max_{\mathbf{s}(t),\boldsymbol{\ell}(t)} \quad \sum_{(i,j)\in\mathcal{A}^2} w_{ij}(t)y_{ij}(t) \tag{7a}$$

$$\text{s.t.} \quad 0 \le s_{ij}(t) \le 1 \quad \forall(i,j) \in \mathcal{A}^2 \tag{7b}$$

$$\sum_{(i,j)\in\mathcal{A}^2} \delta^r_{ij} s_{ij}(t) \le 1 \quad \forall r \in \mathcal{R} \tag{7c}$$

$$\boldsymbol{\ell}(t) \in \mathcal{L}(\mathbf{x}(t)) \tag{7d}$$

$$y_{ij}(t) \le s_{ij}(t)Q_{ij}\ell^{\downarrow}_i(t) \quad \forall(i,j) \in \mathcal{A}^2 \tag{7e}$$

$$y_{ij}(t) \le s_{ij}(t)Q_{ij}\ell^{\uparrow}_j(t) \quad \forall(i,j) \in \mathcal{A}^2 \tag{7f}$$

$$y_{ij}(t) \le x_{ij}(t) \quad \forall(i,j) \in \mathcal{A}^2 \tag{7g}$$

$$y_{ij}(t) \ge 0 \quad \forall(i,j) \in \mathcal{A}^2 \tag{7h}$$

The objective of mathematical program (7) is to maximize the weighted sum of intersection flows. Constraint (7d) restricts the possible lane allocations to $\boldsymbol{\ell}(t) \in \mathcal{L}(\mathbf{x}(t))$, since $\mathcal{L}(\mathbf{x}(t))$ is the feasible set of lane allocations as previously defined. Since the flow across each intersection is only affected by incoming and outgoing links, the solution can be found by solving decomposing program (7) by every node $n \in \mathcal{N}$. Further observe that if the lane allocation $\boldsymbol{\ell}(t)$ is fixed, then program (7) becomes a linear program with decision variables $\mathbf{s}(t)$ and $\mathbf{y}(t)$. If $\ell_{ij}(t)$ are restricted to integer values then $\mathcal{L}(\mathbf{x})$ is finite, and after decomposing program (7) per node, the number of possible lane allocations is relatively small. Therefore, solving program (7) is equivalent to solving program (8) for each node with all possible lane allocations at that node. By doing so, $\mathbf{l}(t)$ becomes a constant in program (8) resulting in a linear program.

$$\max_{\mathbf{s}(t),\boldsymbol{\ell}(t)} \quad \sum_{(i,j)\in\Gamma^-(n)\times\Gamma^+(n)} w_{ij}(t)y_{ij}(t) \tag{8a}$$

$$\text{s.t.} \quad 0 \le s_{ij}(t) \le 1 \quad \forall(i,j) \in \Gamma^-(n) \times \Gamma^+(n) \tag{8b}$$

$$\sum_{(i,j)\in\Gamma^-(n)\times\Gamma^+(n)} \delta^r_{ij} s_{ij}(t) \le 1 \quad \forall r \in \mathcal{R}_n \tag{8c}$$

$$y_{ij}(t) \le s_{ij}(t)Q_{ij}\ell^{\downarrow}_i(t) \quad \forall(i,j) \in \Gamma^-(n) \times \Gamma^+(n) \tag{8d}$$

$$y_{ij}(t) \le s_{ij}(t)Q_{ij}\ell^{\uparrow}_j(t) \quad \forall(i,j) \in \Gamma^-(n) \times \Gamma^+(n) \tag{8e}$$

$$y_{ij}(t) \le x_{ij}(t) \quad \forall(i,j) \in \Gamma^-(n) \times \Gamma^+(n) \tag{8f}$$

$$y_{ij}(t) \ge 0 \quad \forall(i,j) \in \Gamma^-(n) \times \Gamma^+(n) \tag{8g}$$

where $\mathcal{R}_n$ is the set of conflict regions at intersection n. Let $\mathbf{s}(t)$ be the array of intersection time allocations, and let $\mathcal{S}$ be the feasible region of linear program (7).

3.5. State transitions

Each time step, the queue lengths evolve based on the lane allocation and intersection control, random turning proportions, and demand. A control $u = (\boldsymbol{\ell}, \mathbf{s})$ specifies both the lane allocation and fractions of time allocated to each turning movement. Let $\mathcal{U}(\mathbf{x}) = (\mathcal{L}(\mathbf{x}), \mathcal{S})$ be the *control space*, also known as the set of feasible controls.

At each time step, the number of vehicles exiting the queue for turning movement $(j,k) \in \mathcal{A}^2$ is $\min\{s_{jk}(t)Q_{jk}(t)\ell_{jk}(t), x_{jk}(t)\}$. The number of vehicles entering the queue for (j,k) depends on vehicle route choices. Let $P_{jk}(t) \in [0,1]$ be the proportion of vehicles entering j that join the queue for turning movement (j,k). When vehicles request intersection access through AIM, they must specify their desired turning movement. Although $P_{jk}(t)$ is a random variable, assume it is realized and known at the beginning of each time step through the AIM protocol. Let p_{jk} be the mean of $P_{jk}(t)$. The number of vehicles entering queue (j,k) is $\sum_{i\in\mathcal{A}} \min\{s_{ij}(t)Q_{ij}\ell_{ij}(t), x_{ij}(t)\}P_{jk}(t)$. Combining these gives the queue update equation for turning movement $(j,k) \in \mathcal{A}_{\text{o}} \times \mathcal{A}$ based on conservation of vehicles:

$$\begin{aligned} x_{jk}(t+1) = x_{jk}(t) &- \min\left\{s_{jk}(t)Q_{jk}(t)\ell_{jk}(t), x_{jk}(t)\right\} \\ &+ \sum_{i\in\mathcal{A}} \min\left\{s_{ij}(t)Q_{ij}\ell_{ij}(t), x_{ij}(t)\right\} P_{jk}(t) \end{aligned} \tag{9}$$

Equation (9) applies for queues on ordinary links, and must be modified for source links to include entering demand. Let $D_{ij}(t) \in \mathbb{R}_+$ be the random variable for demand entering the queue for turning movement $(i,j) \in \mathcal{A}_{\text{r}} \times \mathcal{A}$ at time t. Let d_{ij} be the mean of $D_{ij}(t)$ and let $\mathbf{d}$ be the vector of average demands. Then the queue update equation for turning movement $(j,k) \in \mathcal{A}_{\text{r}} \times \mathcal{A}$ is as follows:

$$x_{jk}(t+1) = x_{jk}(t) - \min\left\{s_{jk}(t)Q_{jk}(t)\ell_{jk}(t), x_{jk}(t)\right\} + D_{jk}(t) \tag{10}$$

Once vehicles reach a sink link, they exit the network, so the queue on a sink link is always empty.

3.6. Stability

We wish to find a control policy that stabilizes queue lengths for as many demands as possible. As in Tassiulas and Ephremides (1992) and Varaiya (2013), we first characterize the demands that can be stabilized, then develop a maximum stability control policy. A control policy π defines the action $u = \pi(\mathbf{x}(t))$ taken when the state is $\mathbf{x}(t)$. A policy π is *feasible* if $\pi(\mathbf{x}(t)) \in \mathcal{U}(\mathbf{x}(t))$ for all $\mathbf{x}(t)$.

Definition 3.1: The queue length process $X(t) = \{\mathbf{x}(t)\}$ is *stable* under policy π if there exists a $K < \infty$ such that

$$\lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathrm{E}\left[\sum_{(i,j)\in\mathcal{A}^2} x_{ij}(t)\right] < K \tag{11}$$

Define $|\mathbf{x}(t)| = \sum_{(i,j)\in\mathcal{A}^2} x_{ij}(t)$ to be the L_1 norm, i.e. the sum of the queue lengths at time t.

As in Varaiya (2013) we will characterize the stabilizable demands in terms of the long-term average controls. For any time t, $D_{ij}(t)$ has mean d_{ij}, which is also the long-term average rate of demand. Let f_i be the average flow on link i, defined by

$$f_j = \sum_{i\in\mathcal{A}} f_i p_{ij} \quad i \in \mathcal{A}_{\text{o}} \tag{12a}$$

$$f_j = \sum_{k\in\mathcal{A}} d_{jk} \quad i \in \mathcal{A}_{\text{r}} \tag{12b}$$

Let $\mathbf{f}$ be a vector of flows, and let $\mathbf{P}$ be the matrix of average turning proportions. By Proposition 1 of Varaiya (2013) there exists an unique flow vector $\mathbf{f}$ satisfying (12a) and (12b) such that $\mathbf{f} = \mathbf{dP}$.

At each time t a control $u(t)$ is chosen which specifies the lane allocations $\boldsymbol{\ell}(t) \in \mathcal{L}(\mathbf{x}(t))$ and intersection time allocations $\mathbf{s}(t) \in \mathcal{S}$. Let $U = \{u(t)\} = (L, S)$ be the network control sequence where

$L = \{\boldsymbol{\ell}(t)\}$ and $S = \{\mathbf{s}(t)\}$ are the sequences of lane allocations and intersection time allocations, respectively. Let $\bar{\boldsymbol{\ell}}$ and $\bar{\mathbf{s}}$ be the average control choices:

$$\bar{\boldsymbol{\ell}} = \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{\ell}(t) \tag{13}$$

$$\bar{\mathbf{s}} = \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbf{s}(t) \tag{14}$$

Definition 3.2: **d** is a stabilizable demand if the associated flow vector **f** satisfies

$$q_{ij}\bar{\ell}_{ij}\bar{s}_{ij} > f_i p_{ij} \tag{15}$$

for all $(i,j) \in \mathcal{A}^2$.

$Q_{ij}\bar{\ell}_{ij}\bar{s}_{ij}$ is the average maximum flow for turning movement (i,j). $f_i p_{ij}$ is the average demand for flow from i to j. If inequality (15) is satisfied, then average demand is less than the average capacity.

Let $co(\mathcal{U})$ be the convex hull of the set of network controls:

$$co(\mathcal{U}) = \left\{ \sum_{u\in\mathcal{U}} \lambda_u u : \lambda_u \geq 0, \sum_{u\in\mathcal{U}} \lambda_u = 1 \right\} \tag{16}$$

Here $co(\mathcal{U})$ is the set of convex combinations of any control $u = (\boldsymbol{\ell}, \mathbf{s})$. As in Varaiya (2013), we use $co(\mathcal{U})$ to characterize the set of stabilizable demands. By Proposition 2 of Varaiya (2013), we observe the following:

Lemma 3.1: *Control $\bar{u} = (\bar{\boldsymbol{\ell}}, \bar{\mathbf{s}}) \in co(\mathcal{U})$ if and only if there exists a control sequence $U = \{u(t) \in \mathcal{U}\}$ satisfying equations* (13) *and* (14).

Let $\mathcal{D}$ be the set of demand vectors such that there exists a $\bar{u} = (\bar{\boldsymbol{\ell}}, \bar{\mathbf{s}}) \in co(\mathcal{U})$ satisfying

$$f_i p_{ij} \leq Q_{ij}\bar{\ell}_{ij}\bar{s}_{ij} \tag{17}$$

Let $\mathcal{D}^0$ denote the interior of $\mathcal{D}$, i.e. the set of demand vectors such that there exists a $\bar{u} = (\bar{\boldsymbol{\ell}}, \bar{\mathbf{s}}) \in co(\mathcal{U})$ satisfying

$$f_i p_{ij} < Q_{ij}\bar{\ell}_{ij}\bar{s}_{ij} \tag{18}$$

Section 3.7 will develop a policy that stabilizes the queue lengths for all $\mathbf{d} \in \mathcal{D}^0$. Stability is not possible if $\mathbf{d} \notin \mathcal{D}$:

Proposition 3.1: *If* $\mathbf{d} \notin \mathcal{D}$, *there is no stabilizing control.*

Proof: Let $\mathbf{f} = \mathbf{dP}$ be the flow vector associated with $\mathbf{d}$. Let λ_u^* be an optimal solution to the linear program

$$\min \quad \sum_{u\in\mathcal{U}} \lambda_u \tag{19a}$$

$$\text{s.t.} \quad f_i p_{ij} \leq \sum_{u\in\mathcal{U}} \left(\lambda_u Q_{ij} \ell_i^{\downarrow} s_{ij}\right) \quad \forall (i,j) \in \mathcal{A}^2 \tag{19b}$$

$$f_i p_{ij} \leq \sum_{u\in\mathcal{U}} \left(\lambda_u q_{ij} \ell_j^{\uparrow} s_{ij}\right) \quad \forall (i,j) \in \mathcal{A}^2 \tag{19c}$$

$$\lambda_u \geq 0 \quad \forall u \in \mathcal{U} \tag{19d}$$

where $\boldsymbol{\ell}$ and $\mathbf{s}$ in equation (19b) are defined by $u = (\boldsymbol{\ell}, \mathbf{s})$. Since $\mathcal{D}$ is defined in terms of equation (17), if $\mathbf{d} \notin \mathcal{D}$ then $\sum_{u\in\mathcal{U}} \lambda_u^* > 1$. By Lemma 3.1, any policy requiring $\sum_{u\in\mathcal{U}} \lambda_u^* > 1$ is not feasible. ■

Note that linear program (19) may have uncountably infinite variables due to the feasible intersection control space $\mathcal{S}$, but this does not present an issue for the proof. Infinite linear programs have been studied in the literature (e.g. Shivakumar, Sivakumar, and Zhang 2016) and the existence of a solution is sufficient for the proof of Proposition 3.1. Actual implementation of the max-pressure policy uses the finite program (8).

The following result will be used for numerically characterizing the stability region.

Proposition 3.2: *Consider two demand rate vectors* $\mathbf{d}$ *and* $\mathbf{d}'$. *Suppose* $\mathbf{d}' > \mathbf{d}$.

(a) *If* $\mathbf{d} \notin \mathcal{D}^0$, *then* $\mathbf{d}' \notin \mathcal{D}^0$.
(b) *If* $\mathbf{d}' \in \mathcal{D}^0$, *then* $\mathbf{d} \in \mathcal{D}^0$.

Proof: Follows from equation (18). ■

3.7. Max-pressure control

We now define the max-pressure policy π^* that stabilizes queue lengths whenever the average demand vector is $\mathbf{d} \in \mathcal{D}^0$. First, define a weight $w_{ij}(t)$ per turning movement (i,j). As in Tassiulas and Ephremides (1992) and Varaiya (2013), this weight increases with the queue on (i,j) and decreases with the queues on downstream links:

$$w_{ij}(t) = x_{ij}(t) - \sum_{k \in \Gamma^+(j)} p_{jk} x_{jk}(t) \tag{20}$$

This weight is the *pressure* to move vehicles from i to j, and the pressure is based on the difference in queue lengths. The max-pressure policy chooses the control which maximizes the relieved pressure:

$$\pi^*(\mathbf{x}(t)) = \max_{u \in \mathcal{U}(\mathbf{x}(t))} \left\{ \sum_{(i,j) \in \mathcal{A}^2} Q_{ij} s_{ij}(t) \ell_{ij}(t) w_{ij}(t) \right\} \tag{21}$$

$\pi^*(\mathbf{x}(t))$ can be found by solving linear program (7) for all feasible lane allocations $\mathcal{L}(\mathbf{x}(t))$. Since $\ell_i \in \mathbb{Z}_+$ and $0 \leq \ell_i \leq \tilde{\ell}_i$ for every link i, $\mathcal{L}(\mathbf{x})$ is finite. Since the lane allocation can vary for upstream and downstream ends of each link, the solution method may be separated by intersection, i.e. solving linear program (8) for all feasible lane allocations per intersection. The solutions per intersection can be solved in parallel and combined to find π^*.

Proposition 3.3: *The policy* π^* *stabilizes queue lengths whenever the average demand vector is* $\mathbf{d} \in \mathcal{D}^0$.

Proof: The proof is almost identical to the proof of Theorem 2 of Varaiya (2013)[1] by replacing $Q_{ij} s_{ij}(t)$ with $Q_{ij} s_{ij}(t) \ell_{ij}(t)$. Note also that we assume Q_{ij} is constant for AVs, whereas in Varaiya's proof the saturation flows are stochastic. ■

3.8. Traffic signals with fixed lanes

Notice that as in Varaiya (2013), $\mathcal{D}^0$ is defined implicitly by inequality (18). Intuitively, $\mathcal{D}^0$ depends on the control space $\mathcal{U}(\mathbf{x})$. AIM and DLR both greatly extend the control space beyond the traffic signals studied by Varaiya (2013). AIM admits simultaneous conflicting combinations of vehicle movements by coordinating the timing of individual vehicles, and DLR changes the numbers of lanes available for outgoing and incoming flows. We modify the control space $\mathcal{U}(\mathbf{x})$ to create $\mathcal{U}_{\text{signal}}(\mathbf{x})$ to illustrate the differences between traffic signals and DLR with AIM.

For traffic signals, DLR is not used, so $\ell_i^{\uparrow}(t) = \ell_i^{\downarrow}(t) = \tilde{\ell}_i$ for all links i. Traffic signals also choose a single phase per time step, so

$$s_{ij}(t) \in \{0, 1\} \tag{22}$$

for all (i, j). Note that constraint (7b) still holds. Although Q_{ij} is realized, it is not known to a traffic signal controller.

Therefore, combining constraint (22) with constraint (7b) prevents multiple conflicting turning movements being activated during a single phase. Constraint (22) results in a finite $\mathcal{S}$, which defines the feasible phases available to the traffic signal. Then program (7) can be simplified to choosing the phase at each intersection that maximizes the relieved pressure.

3.9. Discussion

We would like to clarify that the analytical results presented here rely on a simplified model of traffic flow. As with previous results on max-pressure control (e.g. Varaiya 2013), the traffic flow model uses point queues. Queue spillback is not included, and backwards-moving shockwaves do not obstruct flow within links. Therefore, the stability results are limited due to the assumptions of the traffic flow model. The stability results further assume that the turning proportions are fixed. In reality, vehicles adjust their route choices to reduce their travel disutility, such as the user equilibrium principle. The purpose of Section 5 is to apply the nice analytical properties from this simplified model to a more realistic model of traffic flow and vehicle behavior.

4. Characterizing the stability region

Section 3.8 shows that simple restrictions on the control space yields a signal control model. Of course, the signal control model has a similar throughput-optimal policy (Varaiya 2013), but the set of stabilizable demands $\mathcal{D}^0$ may be different. A comparison of $\mathcal{D}^0$ for DLR, AIM, and signals would show the potential capacity improvements from installing DLR and AIM technologies.

Equation (18), which is used to define $\mathcal{D}^0$, does not give a clear method for enumerating the boundaries of $\mathcal{D}^0$. The stability region is *characterized* in the sense that $\mathbf{d} \in \mathcal{D}^0$ if there exists a $\bar{\mathbf{s}}$ and $\bar{\ell}$ satisfying equation (18). Checking this existence is required for enumeration. As specified by equation (18), we must determine whether there exists an average control $\bar{u}$ such that the average flow proportions are less than the average intersection flow permitted by $\bar{u}$. Equivalently, for given $\mathbf{d}$ and $\mathbf{P}$, we must solve linear program (19) to determine whether $\sum_{u \in \mathcal{U}} \lambda_u \leq 1$. However, $\mathcal{U}(\mathbf{x})$ may be uncountably infinite for AIM.

Although calculating (18) does not seem practical, it is nevertheless possible to find the optimal policy π^* relatively easily by solving program (8) for all possible lane allocations. Therefore, to find the boundaries of $\mathcal{D}^0$, we use simulation. Choose a duration T, then simulate the evolution of queue lengths from $t = 0$ to T. At each time step, choose $u = \pi^*(\mathbf{x}(t))$, then calculate $\mathbf{x}(t+1)$ according to equation (10) based on the demand.

4.1. Relation between signals and AIM

Consider signals and AIM without DLR. As we will show below, it is important for traffic signal lost time to be included as a reduction in the exogenous saturation flow parameter. Let $\mathcal{U}_{\text{aim}}$ be the set of feasible controls for AIM without DLR (setting $\ell_i(t) = \tilde{\ell}_i$ for all $i \in \mathcal{A}$). Let $\mathcal{D}^0_{\text{signal}}$ and $\mathcal{D}^0_{\text{aim}}$ be the region of stable demands for signals and AIM without DLR, respectively.

Proposition 4.1: $\mathcal{D}^0_{\text{signal}} = \mathcal{D}^0_{\text{aim}}$.

Proof: Consider any $\bar{\mathbf{s}}$ obtainable from AIM. Then $\bar{\mathbf{s}}$ satisfies

$$\sum_{(i,j)\in\mathcal{A}^2} \delta^r_{ij}\bar{s}_{ij} \le 1 \quad \forall r \in \mathcal{R} \tag{23}$$

from constraint (7c). Let $\mathcal{S}_{\text{signal}}$ be the possible intersection control matrices for signals. Each $\mathbf{s} \in \mathcal{S}_{\text{signal}}$ defines a phase.

Since $\mathcal{S}_{\text{signal}}$ is defined by constraints (7c) and (22), $\bar{\mathbf{s}}$ is in the convex hull of $\mathcal{S}_{\text{signal}}$. Then there exists some $\lambda_{\mathbf{s}}$ satisfying

$$\bar{\mathbf{s}} = \sum_{\mathbf{s}\in\mathcal{S}_{\text{signal}}} \lambda_{\mathbf{s}}\mathbf{s} \tag{24a}$$

$$\sum_{\mathbf{s}\in\mathcal{S}_{\text{signal}}} \lambda_{\mathbf{s}} = 1 \tag{24b}$$

Over a sufficiently long time period, we can obtain $\bar{\mathbf{s}}$ by choosing phase $\mathbf{s} \in \mathcal{S}_{\text{signal}}$ for $\lambda_{\mathbf{s}}$ proportion of the time.

Then $\bar{\mathbf{s}}$ is also obtainable through traffic signals, which results in $\mathcal{D}^0_{\text{aim}} = \mathcal{D}^0_{\text{signal}}$ by equation (18). ■

Of course, signals add additional overhead in terms of startup and clearance lost time, and queues of vehicles cannot immediately move due to non-infinite backwards wave speeds. AVs under the AIM control need not have the same overhead because they can coordinate their accelerations and startup timing through vehicle-to-intersection communications.

Therefore, Proposition 4.1 shows that these effects must be included in the exogenous specification of the saturation flow parameters because they are not endogenous to the stochastic queueing model itself.

Proposition 4.1 is somewhat unintuitive, and holds for two main reasons. First, lost time from signal phases is likely to result in lower saturation flows for signals. Second, even with $\mathcal{D}^0_{\text{signal}} = \mathcal{D}^0_{\text{aim}}$, using signals will likely result in greater variance in the queue length for any given turning movements because signal phases satisfy constraint (22). Consequently, outflow from long queues will be further reduced by a non-infinite backwards wave speed and signals might cause queue spillback whereas AIM would not. Neither of these effects are modeled in Section 3. Still, Proposition 4.1 is an important result for analyzing the change in stability region from signals to AIM without DLR. The stability region of signals will be proportionally reduced from the stability region of AIM based on the fraction of each time step lost to startup delay and clearance intervals.

4.2. Detecting stabilizable demands

A more interesting part of this simulation method is detecting an unstabilizable demand. After the termination of simulation, we have recorded the queue lengths $\mathbf{x}(t)$ at every time step t. Since the policy is decentralized, we investigated $\mathcal{D}^0$ for a single intersection. Nevertheless, the same method could easily be applied to arbitrary networks. We considered a standard four-approach intersection, with approaches from the north, south, east, and west. Each approach had two lanes per approach, and all outgoing links also had two lanes. Conflict regions were determined as in Levin and Boyles (2015), illustrated in Figure 1. We chose $Q_{ij} = 1200$ vph. Turning proportions were distributed on average as 70% through, 20% right-turn, and 10% left-turn movements, but actual turning proportions at any time step varied depending on the Poisson distribution sampling for demand per turning movement. We used IBM's CPLEX solver (version 12.6.0) to find the optimal solution to program (7). For a 2hr simulation with a 15s time step, computation times were on average 68.9s on an Intel i7–7700 CPU with 16GB of memory. Figure 2 plots the sum of the queue lengths $|\mathbf{x}(t)|$ for two demand scenarios for $T = 3600$ s with a time step of 15 s using a Poisson distribution for demand.

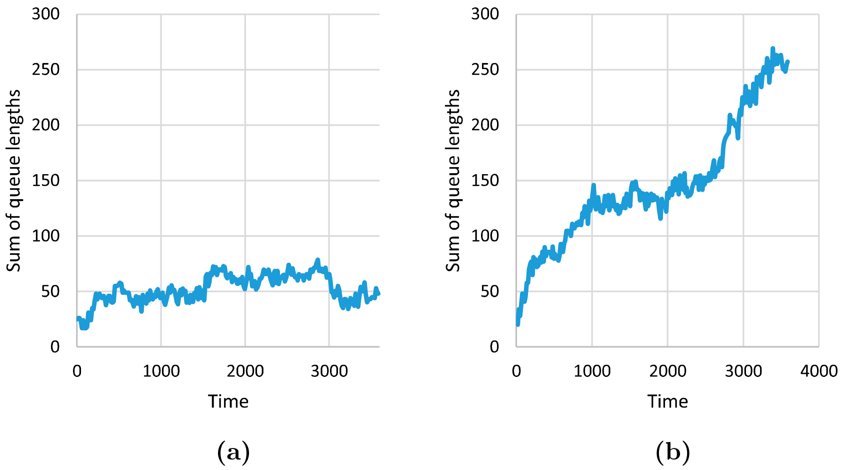

Figure 2. Queue lengths over time for one simulation using **(a)** 1500vph demand per approach and **(b)** 1800vph demand per approach. Each approach had 70% straight, 20% right-turn, and 10% left-turn movements with a Poisson distribution for demand.(a) 1500 vph demand, (b) 1800 vph demand.

From Figure 2 it is clearly obvious from a sufficiently long time period whether a demand vector is stabilizable or not. In Figure 2(a), the queue lengths increase sharply during the middle of the simulation, but return to around 20 several times. On the other hand, in Figure 2(b), the queue lengths are on average (but not monotonically) increasing with time. Notice that in Figure 2(a), the queue lengths did not return to 0, but the queue lengths nevertheless satisfy Definition 3.1 as the average queue length is bounded by some K.

Evaluating Definition 3.1 would require running a simulation for an infinite time horizon. An alternative identification comes from realizing that demand is stabilizable if the queue length process $\mathbf{x}(t)$ is positive recurrent. Formally, the queue length process is positive recurrent if for every $\mathbf{x}(t)$, there exists a τ such that the probability that $\mathbf{x}(t+\tau) = \mathbf{x}(t)$ given $\mathbf{x}(t)$ is non-zero. This definition is more feasible to check within a finite simulation time. Over a simulation duration of T, we record the sum of the queue lengths at τ, $|\mathbf{x}(\tau)|$. Then, we check whether the sum of the queue lengths ever recurs to within $(1+\epsilon)|\mathbf{x}(\tau)|$ in the last τ time of simulation. If so, we mark the demand rate as stabilizable. It is clear from Figure 2(b) that demands that are far from the stability region will not satisfy this check when ϵ is chosen to be a reasonably small number. We chose $\epsilon = 0.1$ to avoid rejecting stabilizable demands for which the simulated queue length at 900s was on the low end of the normal variation.

Notice that equation (18), which defines $\mathcal{D}^0$, depends only on the average rate of entering demand. The *distribution* of the entering demand is not relevant. Of course, the distribution will affect the average queue length and delay, but does not affect whether a demand rate vector $\mathbf{d}$ is stabilizable. Therefore, we can restrict our attention to deterministic demand.

Following the preceding discussion, we can now present a general method for enumerating $\mathcal{D}^0$. Recall that $\mathcal{A}_\mathrm{r}$ is the set of links with external demand entering the network. Let $\Sigma d_i = \sum_{j\in\mathcal{A}} d_{ij}$ be the total rate of demand entering approach i. Divide $\mathcal{A}_\mathrm{r}$ into $\mathcal{A}_\mathrm{r}^\mathrm{fix} \subsetneq \mathcal{A}_\mathrm{r}$ with fixed demand and $\mathcal{A}_\mathrm{r}^\mathrm{var} \subsetneq \mathcal{A}_\mathrm{r}$ with demand that varies together. Fix Σd_i for all $i \in \mathcal{A}_\mathrm{r}^\mathrm{fix}$. Since $\bar{\Delta}x$ should be zero or monotone increasing, use a binary search on $\mathcal{A}_\mathrm{r}^\mathrm{var}$ (with the demand on each $i \in \mathcal{A}_\mathrm{r}^\mathrm{var}$ varying together).

4.3. Single-intersection stability region

We used the method in Section 4.2 to graph the stability region for the network with one standard four-approach intersection described in Section 4.2. We used $\tau = 900$ s and $T = 7200$ s with a time step of 15 s. To reduce the number of variables in the binary search approach, we first assumed that

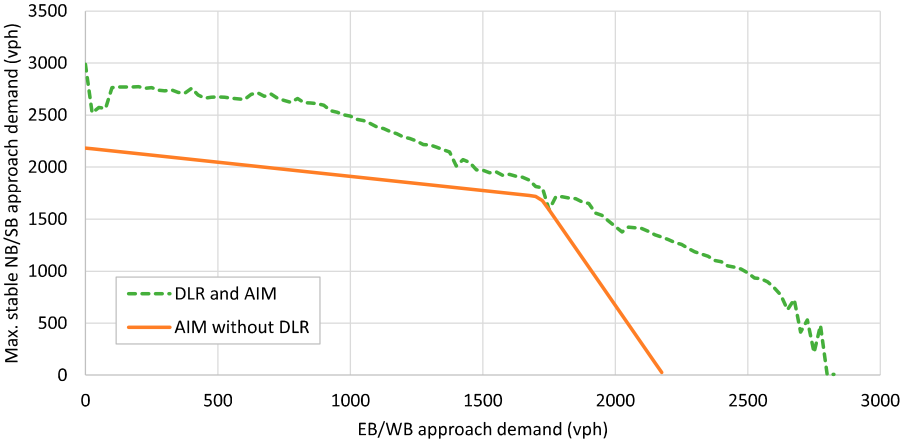

Figure 3. Maximum stable northbound and southbound demand for given eastbound and westbound demand for standard four-approach intersection with different control technologies.

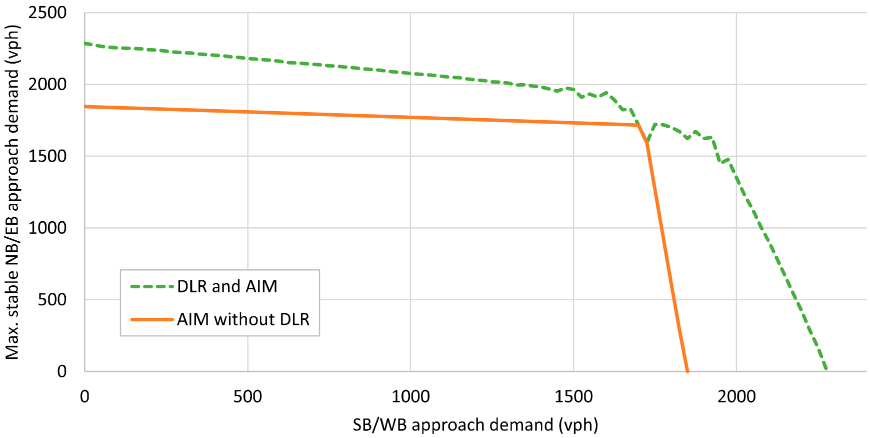

Figure 4. Maximum stable northbound and eastbound demand for given southbound and westbound demand for standard four-approach intersection with different control technologies.

the north-south approaches and the east-west approaches have the same demand. Figure 3 plots the maximum northbound and southbound demand rate for which queue lengths were stable, given a fixed eastbound and westbound demand rate. Figure 4 does the same for northbound/eastbound and southbound/westbound demand, respectively. The demand variations in Figure 4 conflict more, and are more interesting when considered with DLR. We considered three intersection control options: AIM with DLR, AIM without DLR, and signals without DLR. (Signals with DLR is unlikely to occur due to the technological and protocol requirements of DLR.) Figures 3 and 4 leave out the stability region for signals, which is equal to the stability region for AIM without DLR by Proposition 4.1.

For AIM without DLR, the maximum NB/SB demand (with 0 EB/WB demand) was around 2182.6 vph, but the maximum NB/EB demand (with 0 SB/WB demand) was around 1847.5 vph. That discrepancy is because NB/SB approaches have fewer conflicts (only left-turning vehicles) than the NB/EB approaches, which conflict in their through movements also. In addition, the stability region appears to come to a sharp extreme point in both Figures 3 and 4. This is due to the intersection geometry. Figure 4 has an especially sharp cutoff because NB/EB demand significantly conflict with each other as well as with SB/WB demand.

AIM with DLR exhibits surprising variability in Figures 3 – the maximum stable NB/SB demand is not monotone decreasing with respect to EB/WB demand. The variation is due to DLR being able to move larger numbers of vehicles through the intersection during any one time step by modifying lane direction. Consequently, the initial queue length sample $|\mathbf{x}(\tau)|$ was sometimes abnormally low. Nevertheless, the overall pattern demonstrates the expected result. The stability region from DLR is significantly greater than the stability region from AIM alone. DLR is also more able to adjust to asymmetric demand. The differences between the stability regions of DLR and AIM are largest when all of the demand is from either the NB/SB or EB/WB approaches (Figure 3) or from either the NB/EB or SB/WB approaches (Figure 4). Therefore, DLR is likely to be particularly beneficial during the AM and PM peak hours when most traffic is moving either towards or away from the central business districts.

5. Simulation-based dynamic traffic assignment

The formulation and proof of the max-pressure control in Section 3 require fairly restrictive assumptions on the traffic network and traffic flow behavior that are necessary to prove the maximum stability of π^*. As in Varaiya (2013), we assumed unbounded queues to prove stability and ignored FIFO behavior. In reality, queues have a finite maximum length due to physical space constraints, and after reaching that length, will begin to spill back onto upstream links. Queue spillback can be a major contributor to high travel times and route choice paradoxes (Daganzo 1998). FIFO may restrict exiting flow, reducing the efficiency of a policy. The max-pressure control formulation also assumes that routing proportions are fixed. In reality, vehicles change routes in response to travel times, and Smith (1979) demonstrated that optimizing signal timings for current proportions and capacities could increase congestion. To explore the practical utility of the max-pressure control developed in Section 3, we adapt π^* to the simulation-based DTA model developed by Levin and Boyles (2016). We begin by briefly describing the simulation-based DTA model in Section 5.1. The novel max-pressure heuristic is then defined in Sections 5.2 and 5.3.

5.1. AIM and DLR in the cell transmission model

For dynamic network loading we use the cell transmission model (CTM) developed by Levin and Boyles (2016) for modeling DLR and AIM. Each link $i \in \mathcal{A}$ is divided into a sequence of *cells*. Let $\mathcal{C}$ be the set of all cells, and let $\mathcal{C}_i \subseteq \mathcal{C}$ be the set of cells on link $i \in \mathcal{A}$. Therefore, vehicles traveling at the free flow speed will move one cell per time step. Each cell c has some number of lanes $\ell_c(t)$ which may vary with time t. Each cell c in a link $i \in \mathcal{A}_{\text{dlr}}$ also has a parallel, opposite direction cell, which we denote $c^{\leftarrow}$.

CTM determines vehicle movement between cells based on the lane allocation (Levin and Boyles 2016). Let $x_c(t)$ be the number of vehicles in cell c at time t. Cell occupancy evolves via conservation of flow:

$$x_c(t+1) = x_c(t) - y_{c,c+1}(t) + y_{c-1,c}(t) \tag{25}$$

where $c+1$ denotes the cell after cell c. $y_{c,c+1}(t)$ is the number of vehicles moving from cell c to cell $c+1$ at time t:

$$y_{c,c+1}(t) = \min\left\{x_c(t), Q_c\Delta t\ell_c(t), \frac{w_{c+1}}{u^{\text{f}}_{c+1}}\left(k_{\text{j}}u^{\text{f}}_{c+1}\Delta t\ell_{c+1}(t) - x_{c+1}(t)\right)\right\} \tag{26}$$

where u^{f}_c is the free flow speed, w_c is the congested wave speed, and Q_c is the capacity of cell c. k_{j} is the jam density.

We refer the reader to Levin and Boyles (2016) for a more detailed discussion of lane-changing behavior within this CTM. The lane allocation is constrained so that vehicles can be forced to change lanes at most once per time step:

$$|\ell_c(t) - \ell_c(t+1)| \leq 1 \tag{27a}$$

$$|\ell_c(t) - \ell_{c+1}(t+1)| \leq 1 \tag{27b}$$

We also ensure sufficient space for the cell occupancy:

$$k_{\mathrm{j}} u_i^{\mathrm{f}} \Delta t \ell_c(t) \geq x_c(t) \tag{28a}$$

$$\ell_c(t) = \ell_{c^{\leftarrow}}(t) = \tilde{\ell}_c + \tilde{\ell}_{c^{\leftarrow}} \tag{28b}$$

where $\tilde{\ell}_c$ is the number of lanes allocated to cell c in the absence of lane reversals.

To model AIM in simulation-based DTA, we use the aforementioned conflict region model but with discrete vehicles. Let $\mathcal{V}_{ij}(t)$ be the set of vehicles waiting for intersection access from link i to link j at time t, and let $\gamma_v(t)\imath\{0,1\}$ indicate whether vehicle v moves through the intersection at t. We wish to execute the max-pressure policy every time step for thousands of intersections in our simulation-based DTA. Therefore, rather than using a MILP to find the optimal combination of vehicles per intersection, we use the greedy heuristic developed by Levin, Fritz, and Boyles (2017). We calculate the efficiency of moving a vehicle from i to j based on the pressure term $w_{ij}(t)$ and the intersection supply the vehicle would consume. We grant reservations to vehicles in order of greatest efficiency.

5.2. Max-pressure control policy

We now modify the max-pressure policy π^* from Section 3.7 for use within the cell transmission model of AIM and DLR. We first define the weight $w_{ij}(t)$ for movement from link i to link j. Recall that for the stochastic queueing model, the weights are defined by equation (20). We define a similar pressure term here based on cell occupancies. Let $x_{cj}(t)$ be the number of vehicles in cell c that want to turn onto link j (so $\sum_{j\in\mathcal{C}_j} x_{cj}(t) = x_c(t)$ is the total cell occupancy). $x_{cj}(t)$ is analogous to $x_{ij}(t)$, except it is defined for occupancy in cell $c \in \mathcal{C}_i$. Then equation (20) can be written as

$$\hat{w}_{ij}(t) = \sum_{c\in\mathcal{C}_i} x_{cj}(t) - \sum_{j\in\Gamma^+(i)} \sum_{k\in\Gamma^+(j)} p_{jk} \sum_{c\in\mathcal{C}_j} x_{ck}(t) \tag{29}$$

for CTM. Of course, $\hat{w}_{ij}(t)$ cannot become as large as $w_{ij}(t)$ because in the stochastic queueing model, queues were unbounded and $w_{ij}(t)$ could become infinitely large as queue lengths grew to infinity. In CTM, queues have a maximum length, which limits the maximum value of $\hat{w}_{ij}(t)$. Still, $\hat{w}_{ij}(t)$ will increase and apply more pressure on movement from i to j as the queue length for (i,j) increases.

In equation (29), the average turning proportions p_{jk} are an extra exogenous parameter to obtain. However, we can convert it to an endogenous parameter by observing that

$$p_{jk} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T} P_{jk}(t) = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T} \sum_{c\in\mathcal{C}_j} \sum_{k\in\Gamma^+(j)} \frac{x_{ck}(t)}{x_c(t)} \tag{30}$$

Then we approximate p_{jk} with the time-specific turning proportions $\hat{p}_{jk}(t)$ defined by

$$\hat{p}_{jk}(t) = \begin{cases} \frac{x_{ck}(t)}{x_c(t)} & x_c(t) > 0 \\ 0 & \text{else} \end{cases} \tag{31}$$

which results in

$$\hat{w}_{ij}(t) = \sum_{c\in\mathcal{C}_i} x_{cj}(t) - \sum_{j\in\Gamma^+(i)} \sum_{c\in\mathcal{C}_j} x_{ck}(t)\hat{p}_{jk}(t) \tag{32}$$

p_{jk} is approximated with the turning proportions at t because those are the current turning proportions of the queue at t. It is hoped that using this approximation will make the pressure more responsive to the actual queued vehicles.

When using the CTM, a major factor in lane allocation is capacity for vehicle movement through the link, not only capacity for movement through intersections. With many cells per link, the possible combinations of lane allocations for individual cells can become quite large, and the optimal cell-specific lane allocation cannot be found within a reasonable computation time. Therefore, we use the strategy of Levin and Boyles (2016) and allocate lanes according to $\ell_i^{\uparrow}(t)$ and $\ell_i^{\downarrow}(t)$. $\ell_i^{\downarrow}(t)$ applies only to the last cell on the link, in effect creating a dynamic turn bay. For any link i, label the cells on i, $\mathcal{C}_i$, as 1 through $|\mathcal{C}_i|$, with cell $c_i = |\mathcal{C}_i|$ being the most downstream cell. Then the cell lane allocation for cell c_i on link i is

$$\ell_{c_i}(t) = \begin{cases} \ell_i^{\downarrow}(t) & \text{if } c_i = |\mathcal{C}_i| \\ \tilde{\ell}_i + \tilde{\ell}_{i^{\leftarrow}} - \ell_{i^{\leftarrow}}^{\downarrow}(t) & \text{if } c_i = 1 \\ \ell_i^{\uparrow}(t) & \text{else} \end{cases} \tag{33}$$

The second case occurs because the downstream allocation of lanes for link i affects the lane allocation of the first cell on link $i^{\leftarrow}$.

Using Equation (33) to define $\ell_i^{\uparrow}(t)$ and $\ell_i^{\downarrow}(t)$, we can rewrite constraint (28a). Consider any cell c on link i. Then the feasible lane allocations are constrained by

$$k_{\mathrm{j}} u_i^{\mathrm{f}} \Delta t \ell_i^{\downarrow}(t) \geq x_{c_i}(t) \quad \text{if} c_i = |\mathcal{C}_i| \tag{34a}$$

$$k_{\mathrm{j}} u_i^{\mathrm{f}} \Delta t \ell_i^{\uparrow}(t) \geq x_{c_i}(t) \quad \text{if} c_i \neq |\mathcal{C}_i| \tag{34b}$$

Using the weights for specific turning movements, we translate the max-pressure problem (21) into the cell transmission model as the following mathematical program:

$$\max_{\boldsymbol{\ell} \in \mathcal{L}(\mathbf{x})} \sum_{(i,j) \in \mathcal{A}^2} w_{ij}(t) \xi_{ij}(t) \tag{35a}$$

$$\text{s.t.} \quad \xi_{ij}(t) = \sum_{v \in \mathcal{V}_{ij}(t)} \gamma_v(t) \quad \forall (i,j) \in \mathcal{A}^2 \tag{35b}$$

$$k_{\mathrm{j}} u_i^{\mathrm{f}} \Delta t \ell_c(t) \geq x_c(t) \quad \forall c \in \mathcal{C} \tag{35c}$$

where $\gamma_v(t)$ in constraint (35b) is defined by vehicle movements from the simulation-based conflict region model, and $\ell_c(t)$ is defined by equation (33). Due to the number of dimensions in $\mathcal{L}(\mathbf{x})$ and the simulation-based constraint (35b), solving program (35) exactly is still unlikely to be practical for large networks. Constraint (35c) is not as easily decentralized because in the cell transmission model (as in reality), the lane allocation for one direction affects the available space for the vehicle queue in the other direction. We therefore develop a heuristic to choose the lane allocation. Essentially, the max-pressure policy for simulation-based DTA calculates pressure terms $w_{ij}(t)$ for all turning movements, then decentralizes intersection flow and lane allocation separately. Intersection flows are calculated per intersection using the greedy heuristic developed by Levin, Fritz, and Boyles (2017). Lane allocation is decentralized at the link level using the heuristic described in Section 5.3.

5.3. Heuristic for lane allocation

Notice that the definition of $\ell_i^{\uparrow}(t)$ and $\ell_i^{\downarrow}(t)$ in equation (33) means that the space with $\ell_i^{\uparrow}(t)$ lanes allocated may overlap with the space with $\ell_{i^{\leftarrow}}^{\uparrow}(t)$ lanes allocated. This overlap was not an issue in the stochastic queueing model due to the lack of density constraints. Therefore, we also require that

$$\ell_i^{\uparrow}(t) + \ell_{i^{\leftarrow}}^{\uparrow}(t) \leq \tilde{\ell}_i + \tilde{\ell}_{i^{\leftarrow}} \tag{36}$$

The CTM developed by Levin and Boyles (2016) has further link-specific restrictions that a vehicle can only be forced to change lanes once per time step:

$$\left|\ell_{ci}(t) - \ell_{(c+1)i}(t)\right| \leq 1 \tag{37a}$$

$$\left|\ell_{ci}(t) - \ell_{ci}(t+1)\right| \leq 1 \tag{37b}$$

Constraints (36) and (37) limit our ability to decentralize the lane allocation by intersection, as is the case in Section 3. Instead, as in Levin and Boyles (2016), we use a heuristic policy that decentralizes lane allocation by link. We use the calculated pressure terms $w_{ij}(t)$ to choose $\ell_i^{\downarrow}(t)$ and $\ell_i^{\uparrow}(t)$, and assign lanes according to equation (33). A feasible lane allocation always exists by Proposition 1 of Levin and Boyles (2016); a simple feasible solution is to set $\ell_{ci}(t+1) = \ell_{ci}(t)$ for every cell c and every link i.

Consider any pair of links i and $i^{\leftarrow}$. We can easily enumerate the possible values of $\ell_i^{\uparrow}(t)$ and $\ell_{i^{\leftarrow}}^{\uparrow}(t)$ based on constraints (34) and (36). For any $\ell_i^{\uparrow}(t)$, calculate

$$\omega_i(t) = \min\left\{\sum_{j\in\Gamma^+(i)} w_{ij}(t)p_{ij}(t), Q_i\ell_i^{\uparrow}(t)\right\} \tag{38}$$

For each pair of links i and $i^{\leftarrow}$, choose the $\ell_i^{\uparrow}(t)$ and $\ell_{i^{\leftarrow}}^{\uparrow}(t)$ that maximizes $\omega_i(t) + \omega_{i^{\leftarrow}}(t)$. $w_{ij}(t)$ increases with respect to the queue length for movement from i to j, and decreases with respect to the queues on j. Since there are multiple downstream links, and vehicles on link i interact with each other and obey first-in-first-out regardless of downstream link, we multiply $w_{ij}(t)$ by the proportion of vehicles turning from i to j. $Q_i\ell_i^{\uparrow}(t)$ is the capacity for much of the link when using equation (33) to assign lanes. Unlike in the stochastic queueing model of Section 3, lane allocation can cause mid-link bottlenecks due to capacity limitations. By finding the lane allocation that maximizes $\omega_i(t) + \omega_{i^{\leftarrow}}(t)$ for each pair of links i and $i^{\leftarrow}$, we essentially seek to maximize the flow through the link.

Next, we consider adding an extra lane to the last (most downstream) cell of i to allow additional flow to exit (which reduces the lanes allocated to the first cell of $i^{\leftarrow}$). This extra lane is similar to a dynamic turning bay, but could also be used to increase the flow for through movements. Recall that the last cell of i is allocated $\ell_i^{\downarrow}(t)$ lanes. If $\ell_i^{\downarrow}(t) > \ell_i^{\uparrow}(t)$, then i has an extra lane for a turning bay. Let $\omega_i^{\downarrow}(t)$ be the estimation of the benefit of the lane allocation on the last cell of link i:

$$\omega_i^{\downarrow}(t) = \min\left\{x_{|\mathcal{C}_i|}(t), Q_i\left(\ell_i^{\downarrow}(t)\right)\right\} + \min\left\{\sum_{j\in\Gamma^-(i^{\leftarrow})} x_{|\mathcal{C}_j|i^{\leftarrow}}(t), Q_{i^{\leftarrow}}\left(\ell_{i^{\leftarrow}}^{\downarrow}(t)\right)\right\} \tag{39}$$

If $\ell_i^{\downarrow}(t) = \ell_i^{\uparrow}(t) + 1$, then that increases the capacity for flow out of i by the capacity per lane, but also decreases the flow of vehicles within $i^{\leftarrow}$ by the capacity per lane. $\omega_i^{\downarrow}(t)$ attempts to estimate the net benefit. We choose the $\ell_i^{\downarrow}(t) \in \{\ell_i^{\uparrow}(t), \ell_i^{\uparrow}(t) + 1\}$ with the greatest $\omega_i^{\downarrow}(t)$.

Equation (39) does not use the pressure weights directly because the number of lanes in the last cell of link i only affects flow out of cell $|\mathcal{C}_i|$ and flow into cell $|\mathcal{C}_i|^{\leftarrow}$ at the current time. Therefore, equation (39) considers the occupancy of cell $|\mathcal{C}_i|$, $x_{|\mathcal{C}_i|}(t)$, as well as the vehicles seeking to enter cell $|\mathcal{C}_i|^{\leftarrow}$ at the current time, $\sum_{j\in\Gamma^-(i^{\leftarrow})} x_{|\mathcal{C}_j|i^{\leftarrow}}(t)$.

6. Numerical results from dynamic traffic assignment

Using the simulation-based DTA model described in Section 5, we perform experiments on a test network and city network to demonstrate the effectiveness of the max-pressure heuristic. As with the max-pressure control of Section 3, the heuristic does not require any knowledge of future demand, whether it be average demand rates or average turning proportions.

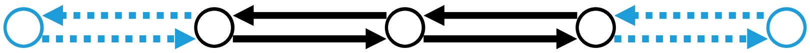

Figure 5. Corridor for benchmark testing of heuristic. The dashed links are centroid connectors.

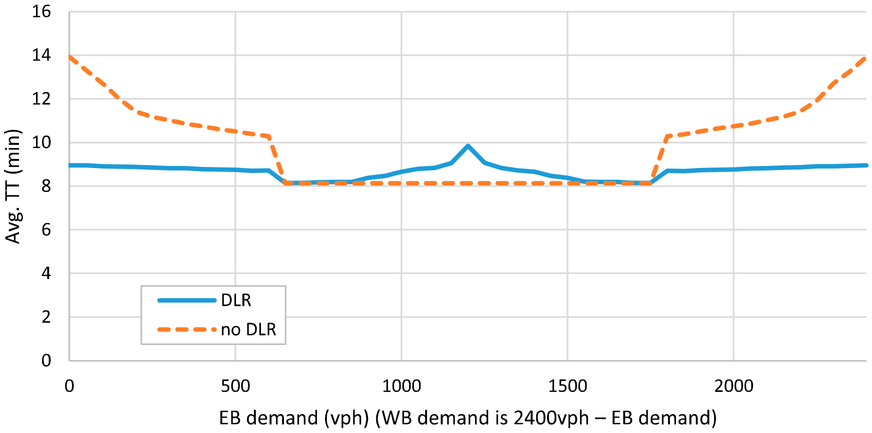

Figure 6. Average travel times for various demand rates. The total demand was always 2400 veh/hr, but the eastbound/westbound split varied. Free flow time is 8 min.

6.1. Test network for dynamic lane reversal

We first study the heuristic on the simple two-way corridor network shown in Figure 5. All intersections were controlled by the max-pressure AIM, and links shown in black were modeled using the cell transmission model with 2 lanes, length 2mi, free flow speed 30mi/hr, congested wave speed 15mi/hr, capacity 800veh/hr, and jam density 264veh/mi. We study a variety of eastbound and westbound demand rates. The purpose of this benchmark is to evaluate whether the heuristic adjusts lane direction properly to account for varying demands. Intersection control was not a major factor due to the lack of turning movement conflicts.

Figure 6 plots average travel times with and without DLR for various demand rates. The total eastbound and westbound demand was 2400 veh/hr, but the split was adjusted between 0 veh/hr eastbound to 2400 veh/hr eastbound. Since each lane has 800 veh/hr capacity, without DLR the capacity per direction was 1600 veh/hr. With DLR, the total capacity (in either direction) of 4800 veh/hr could be redistributed among the two directions of traffic. One hour of demand was simulated, and all vehicles were allowed to exit.

Figure 6 shows that the heuristic was significantly effective at reallocating capacity when demand was highly asymmetric. When all 2400 veh/hr were in one direction, average travel times increased by around 10.0% with DLR compared to 71.3% without DLR. The heuristic performed worst when demand was completely symmetric, with a 20.9% increase in average travel times. When demand is perfectly symmetric, the heuristic reacted to small differences in link occupancies and changed lane directions to compensate. Although frequent changing of lane directions is not an issue with the point-queue model of Section 3, it can cause mid-link bottlenecks and congested regions of flow in the cell transmission model. Nevertheless, this heuristic could be considerably effective during the AM and PM peak hours when most demand is traveling either to or from the central business districts. Therefore, in Section 6.3, we study max-pressure control in the AM peak scenario for the downtown Austin network.

6.2. Stability region analysis

We next study the stability region on the same single-intersection network as in Section 4, except using CTM for the traffic flow model. All links had 2 mile lengths, and were connected to a centroid which

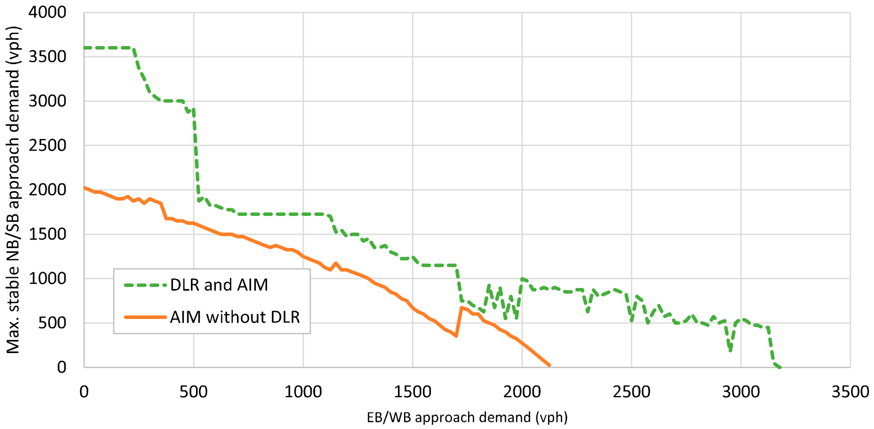

Figure 7. Maximum stable northbound and southbound demand for given eastbound and westbound demand for standard four-approach intersection with different control technologies using CTM.

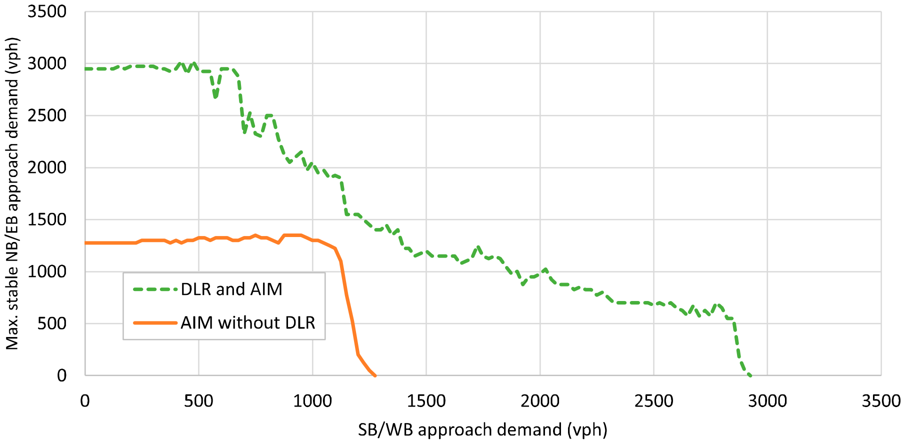

Figure 8. Maximum stable northbound and eastbound demand for given southbound and westbound demand for standard four-approach intersection with different control technologies using CTM.

could hold arbitrarily long queues. Each link had 2 lanes, and capacity was 1200 veh/hr per lane. The free flow speed was set at 30 mi/hr and the congested wave speed at 15 mi/hr. As in Section 4.3, 2 hours of simulation, including continuous demand generation, was used in experiments. We used the method in Section 4.2 to test whether a combination of demands could be stabilized. Figures 7 and 8 are directly comparable to Figures 7 and 8, respectively.

Overall, the results are much noisier than in Figures 3 and 4. That is because the cell transmission model, with a finite congested wave speed and queue spillback, creates much more interesting traffic flow dynamics. The stability was based on whether a queue length at $t = 900$ s of simulation was observed to recur in the last 900s of simulation. Due to the variability in traffic flow, the queue length was unusually low at $t = 900$ s for some demands, and higher for others. Regardless, Figures 7 and 8 show the same general pattern as Figures 3 and 4. DLR is observed to significantly increase the stability region. Actually, the effects of DLR on the north/east-bound vs. south/west-bound approach demands is much more dramatic than in Figure 4, and stabilizes up to 3000 veh/hr demand. On the other hand, AIM without DLR is less effective because the intersection becomes a major bottleneck. Long queues and a finite congested wave speed reduce the efficiency of vehicles exiting the queue.

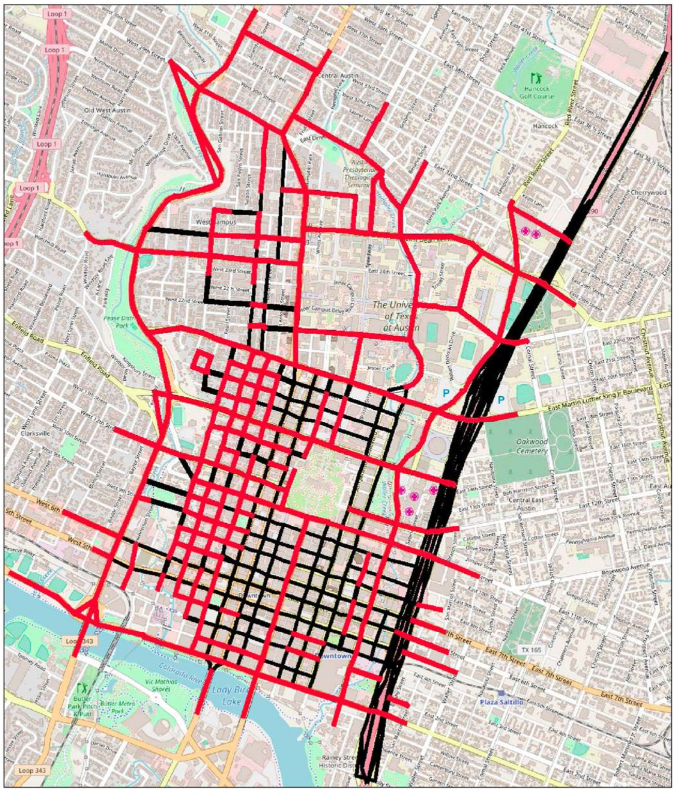

Figure 9. Downtown Austin network. Links shown in red were capable of DLR.

These results clearly show that even in a more realistic cell transmission model-based simulation, DLR greatly increases the stability region. During the AM and PM peak periods, in which most traffic is heading towards or away from the central business districts, respectively, DLR could greatly improve throughput. To study the throughput benefits, we next present results from a network calibrated for the morning peak period.

6.3. Downtown Austin network

To demonstrate the improvements of max-pressure and DLR in a realistic setting, we used the downtown Austin network which was calibrated to match observations by the Network Modeling Center. This network has 171 zones, 546 intersections, 1247 links, and 62,836 trips over 2 hours during the AM peak. The network is shown in Figure 9; links highlighted in red were capable of DLR. Other links were not due to being one-way only, having different lengths or free flow speeds, or having different geometry that would discourage lane reversals. For instance, the I–35 freeway on the east side is divided by its structural design, and reversing lanes would require major structural changes. However, the major arterial corridors are capable of DLR.

We studied four scenarios. The first is a benchmark consisting of controlling intersections with AIM with the first-come-first-served (FCFS) policy. FCFS has been widely used in previous work on AIM (e.g. Dresner and Stone 2004; Fajardo et al. 2011; Li et al. 2013; Levin and Boyles 2016) and is one of the simplest policies both from control, fairness, and implementation perspectives. Nevertheless, FCFS is far from optimal even in simple scenarios (Levin, Boyles, and Patel 2016). Our second scenario is the max-pressure control described in Section 5 without DLR. The third scenario is max-pressure control of both AIM and DLR. To focus on the max-pressure control, capacity increases from AVs are

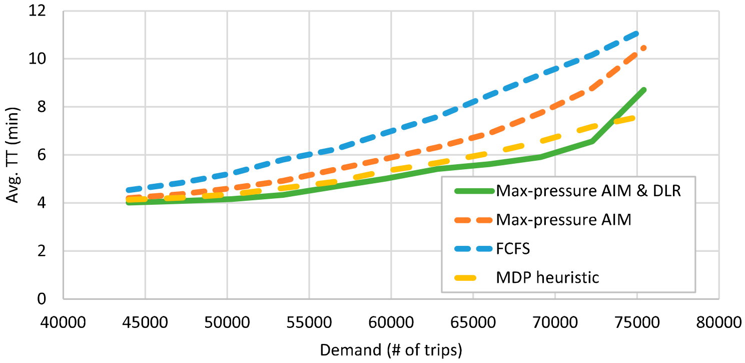

Figure 10. Comparison of max-pressure AIM and DLR, max-pressure AIM without DLR, and first-come-first-served AIM. The 'MDP heuristic' is the heuristic developed by Levin and Boyles (2016).

not included in these scenarios (see Levin and Boyles 2016). For a comparison of traffic signals and FCFS on this network, we refer the reader to Levin, Boyles, and Patel (2016). The fourth (MDP heuristic) implements the lane allocation and intersection control policy of Levin and Boyles (2016).

Figure 10 presents a sensitivity analysis of the average travel times at dynamic user equilibrium with respect to demand. Demand was scaled proportionally based on the calibrated origin-destination matrix. DTA was solved using the method of successive averages to 500 iterations or a cost gap of 0.36 (see Levin et al. 2015), and all vehicles exited. Max-pressure DLR increased the computation time by an average of 17.3% per iteration as compared with FCFS. Scenarios were run on a desktop computer with an Intel i7−7700 CPU at 3.60 GHz with 16 GB of memory.

The max-pressure policy performed around 5% better than the MDP heuristic from Levin and Boyles (2016) for less than 63,000 trips. Between 63,000 and 75,000 trips, greater improvement was observed of up to 9.9% at 69,143 trips. At 75,423 trips, the max-pressure method performed worse because it struggled with gridlock. This is not completely unexpected. Gridlock typically results from queue spillback. The max-pressure policy was developed for a point queue model, which does not include queue spillback.

The maximum observed improvement from max-pressure AIM (without DLR) over FCFS AIM was 18.6%, at 105% of the current demand. The average travel times exhibited an interesting pattern. Improvement from max-pressure AIM was lowest at low demand *and* at high demand. At low demand, congestion was low, so the intersections were less of a bottleneck. At high demand, link capacities became a significant bottleneck, and max-pressure AIM without DLR was less effective at reducing congestion.

AIM and DLR combined was more effective than AIM alone at all demand levels. Max-pressure control of AIM with DLR decreased average travel times by up to 36.8% (at 110% of the current demand). DLR seemed to be most effective at reducing congestion when scaled demand was between 100% and 115% of the calibrated demand. At higher demands, DLR may not have been able to adjust lane directions enough to match capacity to demand. Many roads have 2 lanes in each direction (4 lanes total). Even one vehicle traveling away from downtown during the AM peak required an entire lane to be allocated, resulting in the practical limitation of 3 lanes in one direction. Although max-pressure AIM and DLR can improve congestion, it cannot completely mitigate growth in demand.

7. Conclusions

This paper developed combined max-pressure control for autonomous intersection management (AIM) and dynamic lane reversal (DLR), two new traffic operations technologies for autonomous vehicles (AVs). Due to the complexity of realistic flow models, we first modified a stochastic queueing model to include AIM and DLR. Using the method of Varaiya (2013), we proved that the max-pressure control stabilizes the network (maintains bounded queue lengths) for demands that can be stabilized. Like max-pressure control for traffic signals, the max-pressure policy developed here has several advantages. Max-pressure is *decentralized* by node: the policy depends only on queue lengths on adjacent links, and is therefore easy to compute in parallel for any size network. Max-pressure also adapts to changes demand without requiring predictions or estimations of future demand by responding to queue lengths. Max-pressure requires knowledge of turning ratios, but that information is always necessary for AIM.

Using the max-pressure control, we compared the stability regions (the set of demands that can be stabilized) from traffic signals, AIM, and AIM with DLR. An interesting theoretical result is that within the stochastic queueing model, traffic signals and AIM have the same stability region assuming identical saturation flows. In practice, AIM will increase the saturation flow by reducing lost time and avoiding long queues and queue spillback. On a test network, AIM with DLR was observed to have a significantly larger stability region than AIM with fixed lanes, and the stability region was especially increased for asymmetric demand. Therefore, DLR could be particularly valuable for peak hour congestion.

However, the stochastic queueing model does not include key aspects of traffic flow such as density limitations, mid-link capacity constraints, and first-in-first-out behavior on links. These behaviors are well modeled by micro- and meso-scopic flow models, but such models are analytically complex, and proving the stability of max-pressure control within such models may not be possible. In addition, Levin, Boyles, and Patel (2016) showed that decentralized control could lead to the Daganzo (1998) paradox. Therefore, we applied max-pressure control to simulation-based DTA using the cell transmission model. Due to the additional lane allocation constraints and computational complexity, we used the max-pressure control weights and concepts in two heuristic policies –one for intersection flows, and another for lane allocation. Results on a test network demonstrated that the DLR heuristic was effective at reducing travel times, especially for asymmetric demand. Max-pressure control was also highly effective on the downtown Austin city network (including solving for dynamic user equilibrium) compared with first-come-first-served.

Since AIM and DLR are not practical with (nearly) 100% AV market penetration, there is considerable time and opportunity for future work. Potential extensions include adding system optimal route choice to the max-pressure control *à la* communications packet routing in Tassiulas and Ephremides (1992). Another important question is whether max-pressure control remains throughput-optimal under realistic traffic flow behavior including density constraints, first-in-first-out behavior, and queue spillback. Finally, the AIM and DLR technologies themselves require further development. Although they are theoretically possible with coordinated AV, they have yet to be demonstrated with real AVs. Lane-changing due to DLR could temporarily reduce capacity, and this effect should be estimated to be included. Capacity loss from lane-changing could be included in this model as a reduction in capacity for the next time step. In practice, it will be decades before 100% AV market penetration is achieved, and even afterwards, roads must be shared with pedestrians and cyclists. Modifying the AIM and DLR control protocols to admit non-connected human interactions is necessary for practical use.

Note

1. Varaiya (2013) uses different notation; $Q_{ij}s_{ij}(t)$ is equivalent to $C(l,m)(t+1)S(l,m)(t)$ in Varaiya (2013)'s proof.

Disclosure statement

No potential conflict of interest was reported by the authors.

References

Alonso, Javier, Vicente Milanés, Joshué Pérez, Enrique Onieva, Carlos González, and Teresa De Pedro. 2011. "Autonomous Vehicle Control Systems for Safe Crossroads." *Transportation Research Part C: Emerging Technologies* 19 (6): 1095–1110.

Carlino, Dustin, Stephen D. Boyles, and Peter Stone. 2013. "Auction-based Autonomous Intersection Management. " 16th International IEEE Conference on Intelligent Transportation Systems-(ITSC), 2013, 529–534, IEEE.

Conde Bento, Luis, Ricardo Parafita, Sara Santos, and Urbano Nunes. 2013. "Intelligent Traffic Management at Intersections: Legacy Mode for Vehicles not Equipped with v2v and v2i Communications." 16th International IEEE Conference on Intelligent Transportation Systems-(ITSC), 2013, 726–731, IEEE.

Daganzo, Carlos F. 1994. "The Cell Transmission Model: A Dynamic Representation of Highway Traffic Consistent with the Hydrodynamic Theory." *Transportation Research Part B: Methodological* 28 (4): 269–287.

Daganzo, Carlos F. 1998. "Queue Spillovers in Transportation Networks with a Route Choice." *Transportation Science* 32 (1): 3–11.

Dixit, Vinayak, and Brian Wolshon. 2014. "Evacuation Traffic Dynamics." *Transportation Research Part C: Emerging Technologies* 49: 114–125.

Dresner, Kurt, and Peter Stone. 2004. "Multiagent Traffic Management: A Reservation-based Intersection Control Mechanism." Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 2, 530–537, IEEE Computer Society.

Dresner, Kurt, and Peter Stone. 2006. "Traffic Intersections of the Future." Proceedings of the National Conference on Artificial Intelligence, 21 vols, 1593, Menlo Park, CA; Cambridge, MA; London: AAAI Press, MIT Press; 1999.

Dresner, Kurt M, and Peter Stone. 2007. "Sharing the Road: Autonomous Vehicles Meet Human Drivers." IJCAI, 7 vols. 1263–1268.

Duell, Melissa, Michael W. Levin, Stephen D. Boyles, and S. Travis Waller. 2016. "Impact of Autonomous Vehicles on Traffic Management: Case of Dynamic Lane Reversal." *Transportation Research Record: Journal of the Transportation Research Board* 2567: 87–94.

Fajardo, David, Tsz-Chiu Au, S. Waller, Peter Stone, and David Yang. 2011. "Automated Intersection Control: Performance of Future Innovation Versus Current Traffic Signal Control." *Transportation Research Record: Journal of the Transportation Research Board* 2259: 223–232.

Gregoire, Jean, Emilio Frazzoli, Arnaud De La Fortelle, and Tichakorn Wongpiromsarn. 2014. "Back-pressure Traffic Signal Control with Unknown Routing Rates." *IFAC Proceedings Volumes* 47 (3): 11332–11337.

Hausknecht, Matthew, Tsz-Chiu Au, Peter Stone, David Fajardo, and Travis Waller. 2011. "Dynamic Lane Reversal in Traffic Management." 14th International IEEE Conference on Intelligent Transportation Systems (ITSC), 2011, 1929–1934, IEEE.

Le, Tung, Péter Kovács, Neil Walton, Hai L. Vu, Lachlan L. H. Andrew, and Serge S. P. Hoogendoorn. 2015. "Decentralized Signal Control for Urban Road Networks." *Transportation Research Part C: Emerging Technologies* 58: 431–450.

Le, Tung, Hai L. Vu, Neil Walton, Serge P. Hoogendoorn, Péter Kovács, and Rudesindo N. Queija. 2017. "Utility Optimization Framework for a Distributed Traffic Control of Urban Road Networks." *Transportation Research Part B: Methodological* 105: 539–558.

Levin, Michael W., and Stephen D. Boyles. 2015. "Intersection Auctions and Reservation-based Control in Dynamic Traffic Assignment." *Transportation Research Record: Journal of the Transportation Research Board* 2497: 35–44.

Levin, Michael W., and Stephen D. Boyles. 2016. "A Cell Transmission Model for Dynamic Lane Reversal with Autonomous Vehicles." *Transportation Research Part C: Emerging Technologies* 68: 126–143.

Levin, Michael W., and Stephen D. Boyles. 2016. "A Multiclass Cell Transmission Model for Shared Human and Autonomous Vehicle Roads." *Transportation Research Part C: Emerging Technologies* 62: 103–116.

Levin, Michael W., Stephen D. Boyles, and Rahul Patel. 2016. "Paradoxes of Reservation-based Intersection Controls in Traffic Networks." *Transportation Research Part A: Policy and Practice* 90: 14–25.

Levin, Michael W., Hagen Fritz, and Stephen D. Boyles. 2017. "On Optimizing Reservation-Based Intersection Controls." *IEEE Transactions on Intelligent Transportation Systems* 18 (3): 505–515.

Levin, Michael W., Matt Pool, Travis Owens, Natalia Ruiz Juri, and S. Travis Waller. 2015. "Improving the Convergence of Simulation-based Dynamic Traffic Assignment Methodologies." *Networks and Spatial Economics* 15 (3): 655–676.

Levin, Michael W., and David Rey. 2017. "Conflict-point Formulation of Intersection Control for Autonomous Vehicles." *Transportation Research Part C: Emerging Technologies* 68: 126–143.

Li, Zhixia, Madhav Chitturi, Dongxi Zheng, Andrea Bill, and David Noyce. 2013. "Modeling Reservation-Based Autonomous Intersection Control in VISSIM." *Transportation Research Record: Journal of the Transportation Research Board* 2381: 81–90.

Meng, Qiang, and Hooi Ling Khoo. 2008. "Optimizing Contraflow Scheduling Problem: Model and Algorithm." *Journal of Intelligent Transportation Systems* 12 (3): 126–138.

Qian, Xiangjun, Jean Gregoire, Fabien Moutarde, and Arnaud De La Fortelle. 2014. "Priority-based Coordination of Autonomous and Legacy Vehicles at Intersection." IEEE 17th International Conference on Intelligent Transportation Systems (ITSC), 2014, 1166–1171, IEEE.

Schepperle, Heiko, and Klemens Böhm. 2007. "Agent-based Traffic Control using Auctions." Cooperative Information Agents XI, 119–133, Springer.

Shivakumar, P. N., K. C. Sivakumar, and Yang Zhang. 2016. “Infinite Linear Programming.” In *Infinite Matrices and Their Recent Applications*, 87–92, Springer.

Smith, M. J. 1979. “Traffic Control and Route-choice; a Simple Example.” *Transportation Research Part B: Methodological* 13 (4): 289–294.

Tachet, Remi, Paolo Santi, Stanislav Sobolevsky, Luis Ignacio Reyes-Castro, Emilio Frazzoli, Dirk Helbing, and Carlo Ratti. 2016. “Revisiting Street Intersections Using Slot-based Systems.” *PloS One* 11 (3): e0149607.

Tassiulas, Leandros, and Anthony Ephremides. 1992. “Stability Properties of Constrained Queueing Systems and Scheduling Policies for Maximum Throughput in Multihop Radio Networks.” *Automatic Control, IEEE Transactions on* 37 (12): 1936–1948.

Varaiya, Pravin. 2013. “Max Pressure Control of a Network of Signalized Intersections.” *Transportation Research Part C: Emerging Technologies* 36: 177–195.

Vasirani, Matteo, and Sascha Ossowski. 2012. “A Market-inspired Approach for Intersection Management in Urban Road Traffic Networks.” *Journal of Artificial Intelligence Research* 43: 621–659.

Wang, J. W., H. F. Wang, W. J. Zhang, and W. H. Ip, and Kazuo Furuta. 2013. “Evacuation Planning Based on the Contraflow Technique with Consideration of Evacuation Priorities and Traffic Setup Time.” *Intelligent Transportation Systems, IEEE Transactions on* 14 (1): 480–485.

Xue, Deyi, and Zuomin Dong. 2000. “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.

Zaidi, Ali A., Balázs Kulcsár, and Henk Wymeersch. 2016. “Back-pressure Traffic Signal Control with Fixed and Adaptive Routing for Urban Vehicular Networks.” *IEEE Transactions on Intelligent Transportation Systems* 17 (8): 2134–2143.

Zhang, XinMing, Shi An, and Bing Lei Xie. 2012. “A Cell-Based Regional Evacuation Model with Contra-Flow Lane Deployment.” Advanced Engineering Forum, 5 vols, 20–25, Trans Tech Publ.

Zhang, Rui, Zhijun Li, Cheng Feng, and Shouxu Jiang. 2012. “Traffic Routing Guidance Algorithm based on Backpressure with a Trade-off between User Satisfaction and Traffic Load.” 2012 IEEE Vehicular Technology Conference (VTC Fall), 1–5, IEEE.

Zhou, W. W., P. Livolsi, E. Miska, H. Zhang, and J. Wu, and D. Yang. 1993. “An Intelligent Traffic Responsive Contraflow Lane Control System.” Vehicle Navigation and Information Systems Conference, 1993., Proceedings of the IEEE-IEE, 174–181, IEEE.

Zhu, Feng, and Satish V. Ukkusuri. 2015. “A Linear Programming Formulation for Autonomous Intersection Control Within a Dynamic Traffic Assignment and Connected Vehicle Environment.” *Transportation Research Part C: Emerging Technologies* 55: 363–378.