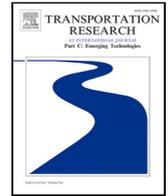


Contents lists available at [ScienceDirect](https://www.sciencedirect.com)

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Optimal decentralized signal control for platooning in connected vehicle networks

The Anh Hoang ^a, Neil Walton ^b, Hai L. Vu ^{a,*}^a Department of Civil Engineering, Monash University, Australia^b Durham University, UK

ARTICLE INFO

Keywords:

Traffic signal control
 Connected vehicles
 Platooning
 Max weight
 Pressure-based control

ABSTRACT

In the last decade, pressure-based schemes such as Back Pressure and Max Weight algorithms have been widely researched and applied for traffic signal control due to their simplicity and proven throughput maximization. In such algorithms, the next chosen signal phase at an intersection in a road network is the one with the highest measured weight, representing the pressure of traffic movements at the intersection, determined based on a single characteristic of the traffic flow or vehicles' state at that intersection. This paper develops a new optimal Max Weight control mechanism to enhance the network throughput and reduce vehicle delays in a network using a concept of platooning enabled by Connected Vehicles (CVs). To this end, we propose a new proven optimal Max Weight control scheme where the weight consists of several features including the platoon delay, as well as the speed and position of vehicles within the platoon. To the best of our knowledge, this work is the first to propose a platoon pressure-based concept considering multiple configurable attributes in formulating the pressure. Furthermore, we provide a rigorous stability proof that ensures the throughput optimality of the proposed control scheme. In addition, we also develop a machine learning procedure in this paper to optimize the weighting parameter of each attribute contributing to the total pressure enabling its seamless deployment in practice. A number of simulation results demonstrate the feasibility of the learning procedure and show that our Max Weight platoon pressure-based scheme outperforms the state-of-the-art and well-known existing pressure-based algorithms.

1. Introduction

Urban traffic congestion is still a pressing problem due to a rapid population growth and mobility demand within the metropolitan areas. In this context, traffic signal control at urban network intersections remain the most widely used tool to effectively manage conflicting traffic flows and mitigate congestion in the network.

There have been many research efforts in the literature dedicated to optimize the urban traffic signal timings in order to efficiently control the traffic network. Nevertheless, the recent advancement in information and communication technology (ICT) enables better connectivity and provides a fresh opportunity to reform and improve the conventional operation of traffic signals and management of the transport networks. More specifically, in the last decade, vehicular communication technologies and infrastructure (i.e., vehicle-to-vehicle V2V, and vehicle-to-infrastructure V2I or simply V2X) have been widely deployed in many countries around the globe. Vehicles equipped with V2X capability, referred to as Connected Vehicles or CVs, can now communicate with each other and with the intersection controllers (or roadside units) to share information in real-time in order to enhance safety

* Corresponding author.

E-mail address: hai.vu@monash.edu (H.L. Vu).

<https://doi.org/10.1016/j.trc.2024.104832>

Received 4 December 2023; Received in revised form 15 July 2024; Accepted 19 August 2024

Available online 2 September 2024

0968-090X/© 2024 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY-NC license (<http://creativecommons.org/licenses/by-nc/4.0/>).

and efficiency of both the individual vehicle and the transport network. As a result, information collected in real-time such as vehicle type, location, speed and acceleration of the approaching or leaving vehicles at an intersection can be used to make more effective control decisions and reduce congestion in the network.

In the literature, numerous studies have attempted to enhance the control algorithm of traffic controllers utilizing real-time collected vehicular data. In particular, [Ubiergo and Jin \(2016\)](#) proposed a hierarchical green driving strategy to smooth vehicles' trajectory, aiming to reduce the overall fuel consumption. In this work, it was assumed that each CV receives an individual speed limit control signal based on its location and current traffic states. On the other hand, [Stebbins et al. \(2017\)](#) improved the Green Light Optimal Speed Advisory system so that vehicles received not only speed but also acceleration guidance to minimize the delay. Furthermore, [Priemer and Friedrich \(2009\)](#) leveraged on V2I technology to propose an optimization framework and solved the problem by using dynamic programming technique. In their work, the intersection manager periodically calculated the signal phase sequence by optimizing the total queue length of vehicles.

Recently, [Emami et al. \(2021\)](#) used the data collected from CVs as input, then formulated traffic signal plans into a linear optimization problem to maximize the throughput of the whole network. In addition, [Wang et al. \(2021\)](#) introduced a two-level traffic control system based on CV technologies. The authors presented an optimization model to optimize throughput for high-demand traffic flows at the corridor level. An adaptive traffic signal control system was then provided at intersection level to reduce vehicle delay.

It is well known that the combined problem of co-optimizing traffic signals and planning vehicle trajectories is very complex ([Nguyen et al., 2021](#)), which requires significant computational resources. Optimized signal timings is essentially creating green waves for as many vehicles to cruise through several intersections as possible together. Consequently, one can develop an approach to form vehicles into platoons as they approach the intersection to improve efficiency rather than relying on the intersection's signal control and green wave to achieve that. For example, [He et al. \(2011\)](#) proposed Platoon-based Arterial Multi-modal Signal Control with online data scheme for connected vehicle networks where vehicles are grouped into platoons. The proposed scheme aimed to find the best signal timing plan to minimize the delay of platoons arriving at intersections. Recently, the work in [Wang et al. \(2020\)](#) also developed a model to optimize arterial signal control based on platoons of connected vehicles by providing speed guidance to pass through the intersection with the least stopping time. Nevertheless, all the traffic signal schemes discussed above were optimized centrally (i.e. centralized control) and hence were not scalable and impractical for deployment over wide areas due to the exponential growth in its computational complexity as the size of the network increases.

To improve scalability, a number of studies instead proposed a distributed algorithm to solve the signal control problem in urban networks ([Priemer and Friedrich, 2009](#); [Al Islam and Hajbabaie, 2017](#); [Liang et al., 2021](#)). One of the promising approaches among the proposed distributed schemes is pressure-based traffic signal control. Theoretically, pressure-based traffic signal control can provide optimal throughput at a network level and does not require any prior knowledge of the traffic demand, and thus is flexible and easily deployable ([Wongpiromsarn et al., 2012](#); [Varaiya, 2013](#)). The application of the pressure-based algorithms initially came from the wireless communication research field, which was introduced in [Tassiulas and Ephremides \(1990\)](#). Since then, its application to traffic control has attracted a huge interest from the transportation research community.

[Table 1](#) provides the summary the existing representative works in traffic signal control utilizing pressure-based algorithms.

There are two variants of pressure-based algorithms in the literature: Back Pressure and Max Weight, where the former was first described by [Tassiulas and Ephremides \(1990\)](#), the latter was introduced by [McKeown et al. \(1999\)](#), both in the telecommunications and data networking research areas. The main difference between Back Pressure and Max Weight algorithms is how the "pressure" of an approaching traffic movement at an intersection, or weight, is defined and calculated. In Back Pressure, the weight of a traffic movement is defined as the difference between the feature of that movement upstream link (i.e. queue length, delay or vehicle spatial distribution in a link) and the value of the corresponding features at the downstream links. In contrast, the weight of a traffic movement in Max Weight is only calculated based on the features of movements at the upstream link of the intersection. For this reason, Max Weight is a simpler mechanism but at the same time less explored due to the difficulties in providing the proof of stability.

In the literature there have been several Back Pressure papers defining the pressure based on the queue length information of the link or traffic movement ([Wongpiromsarn et al., 2012](#); [Varaiya, 2013](#); [Gregoire et al., 2014a,b](#); [Zaidi et al., 2016](#); [Le et al., 2015, 2017](#); [Pumir et al., 2015](#); [Lioris et al., 2016](#)). The authors of [Li and Jabari \(2019\)](#), on the other hand, proposed the position-weighted Back Pressure control policy. The proposed Back Pressure scheme utilized the spatial distribution of vehicles in a link as its defining characteristic. In addition, the work in [Dixit et al. \(2020\)](#) used traffic delay as a feature in their proposed framework, and the work in [Mercader et al. \(2020\)](#) defined pressure using travel time information. Recently, [Liu and Gayah \(2022\)](#) proposed a delay-based Back Pressure scheme based on the mathematical relationship between the number of queuing (standing) vehicles and link's delay.

While Back Pressure control algorithms are very popular in the existing research literature, little attention has been paid to its simpler version, the Max Weight control algorithm, with an exception of [Wu et al. \(2017\)](#). [Wu et al. \(2017\)](#) introduced a delay-based Max Weight for intersection control, where the weight of a traffic movement utilizes the sojourn time of its Head-Of-Line (HOL) vehicle at the intersection. This proposed framework gained better fairness between vehicles; however, only explored the Max Weight scheme for an isolated intersection.

Through the literature review, we found several research gaps that will be addressed in this paper. Firstly, while vehicle platooning in CV network has been explored and shown significant performance improvements ([He et al., 2011](#); [Lioris et al., 2017](#); [Liang et al., 2020](#)), there have been no studies using pressure-based signal control applying for this promising platooning concept. Secondly, there is only a very limited number of studies that used the Max Weight signal control framework for network traffic signal control. Compared to Back Pressure, Max Weight algorithm does not require data from downstream movements at the intersections,

Table 1
Summary of pressure-based signal control studies.

Type	Paper	Pressure measurement	Stability proof	Network consideration	
Back pressure	Wongpiromsarn et al. (2012)	Queue length	✓	✓	
	Varaiya (2013)	Queue length	✓	✓	
	Gregoire et al. (2014a,b)	Queue length	-	✓	
	Le et al. (2015) and Le et al. (2017)	Queue length	✓	✓	
	Pumir et al. (2015)	Queue length	✓	✓	
	Lioris et al. (2016)	Queue length	✓	✓	
	Dixit et al. (2020)	Crowd-sourced traffic delay	-	✓	
	Mercader et al. (2020)	Normalized travel time	-	✓	
				only for work conservative	
		Liu and Gayah (2022)	Vehicle delay	✓	✓
Max weight	Wu et al. (2017)	Vehicle delay	✓	-	
			only for single intersection		
	This work	Queue length, platoons & other local parameters	✓	✓	

hence it is simpler to implement when applying to large-scale networks. Finally, the existing works in the literature of pressure-based algorithms only considered a single feature of the link, vehicle or traffic movement. We expect the signal control performance to enhance further when considering several features together in the design of the pressure-based algorithm. In particular, we will consider multiple characteristics of a platoon for the weight of signal phases. In a sense, our approach can be seen as a generalization of the pressure-based algorithm of individual vehicles. Furthermore, we propose a new algorithm to learn and adapt the parameters of the Max Weight algorithm depending on the traffic flow demand and the resulting platoon characteristics of that demand shaped by the signal control which in turn forms a closed adaptive control system at the road intersections.

To this end, we develop a new platoon-based optimal Max Weight in this work to enhance the network throughput and reduce vehicle delays using platoons with CVs. The contributions of this paper include:

- Propose a Max Weight decentralized signal control algorithm considering vehicle platooning in an urban network of connected vehicles (CVs). In our proposed algorithm, the weight (or pressure) of a traffic movement at an intersection consists of several platoon's features (or characteristics) including its delay, and the speed and position of vehicles in the platoon.
- Develop a cross entropy method as the learning algorithm to optimize the contribution of each feature in the weight of the proposed Max Weight signal control algorithm. The learned algorithm can optimize performance for moderate to low queue sizes while retaining stability for larger queue sizes.
- Prove the stability property of proposed Max Weight signal control algorithm. To the best of our knowledge, this is the first time a *network* road traffic stability result using Max Weight algorithm is obtained.

Furthermore, we provide extensive numerical results via simulation to show the effectiveness of our developed cross entropy method to optimize the proposed Max Weight signal control algorithm. To this end, we extensively compare and show significant improvements in the proposed algorithm over other well-known pressure-based schemes.

The remainder of this paper is organized as follows: Section 2 defines our Learning Optimal Max Weight signal control model. Theoretical results, especially stability property of our Max Weight algorithm is presented in Section 3. Numerical results from the simulation are presented in Section 4, and we conclude the paper in Section 5.

2. Model

2.1. Network model and queuing dynamics

Consider a transport network as a graph that \mathcal{N} is the set of nodes representing intersections. We denote \mathcal{J} as the set of all turning movements, or queues in the network. Furthermore, \mathcal{J}_n is the set of all turning movements at intersection $n \in \mathcal{N}$. Let \mathcal{T} be the set of time steps.

Assume that at the beginning of time step t , each intersection controller of node $n \in \mathcal{N}$ needs to decide whether its corresponding turning movements (or queue) $j \in \mathcal{J}_n$ are activated or not. This decision is indicated by $s_j(t) \in \{0, 1\}$. Let $S(t)$ be the matrix that denotes the activation of all turning movements at all intersections within the network during time step t . Then, we have \mathcal{S} as the set that contains all feasible network control matrices S of intersections in the network. Likewise, for an arbitrary intersection $n \in \mathcal{N}$, $S_n(t)$ and \mathcal{S}_n indicate its turning movements matrix at time t and set of all feasible control matrices in this intersection respectively.

We let $\langle \mathcal{S} \rangle$ be the convex closure of \mathcal{S} . Notice this gives the set of services achievable when we allow for randomness in service. We define \mathcal{C} to be the interior of $\langle \mathcal{S} \rangle$. We call \mathcal{C} the capacity set of our network.

We let $A_j(t)$ denote the number of arrivals at queue j by time t . We let $D_j(t)$ be the number of vehicles that have departed from queue j by time t . Both $A_j(t)$ and $D_j(t)$ are non-decreasing processes for $t \in \mathcal{T}$. We let $Q_j(t)$ be the queue size at time t . From this, it is clear that $Q_j(t)$ should be a non-negative process satisfying

$$Q_j(t) = Q_j(0) + A_j(t) - D_j(t), \quad j \in \mathcal{J}. \quad (1)$$

Arrivals at queue j consist of both external arrivals, that is vehicles joining the queue from outside the network, and internal arrivals, that is vehicles that have moved from another junction within the road network. We let $E_j(t)$ be the total number of external arrivals to queue j by time t . We assume that $E_j(t) - E_j(t-1)$ is non-negative independent identically decentralized with bounded variance and mean a_j . We let $\Phi_{i,j}(s)$ indicate if the s th vehicle to depart queue $i \in \mathcal{J}$ next joins queue $j \in \mathcal{J} \cup \{\emptyset\}$. Specifically, when the s th job to depart queue i next joins queue j we have that $\Phi_{i,j}(s) = 1$ and $\Phi_{i,j'}(s) = 0 \forall j' \neq j$. We assume that $(\Phi_{i,j}(s) : i \in \mathcal{J}, j \in \mathcal{J} \cup \{\emptyset\})$ are independent identically decentralized random variables with mean given by $(P_{i,j} : i \in \mathcal{J}, j \in \mathcal{J} \cup \{\emptyset\})$. Here $j = \emptyset$ indicates that the vehicle departs the network. Given the notation as just described the total number of arrivals to queue j can be expressed as

$$A_j(t) = E_j(t) + \sum_{i \in \mathcal{J}} \sum_{s=1}^{D_j(t)} \Phi_{i,j}(s). \quad (2)$$

We let $\mathbf{a} = (a_j : j \in \mathcal{J})$ be the exogenous arrival rate of vehicles at each queue. That is the mean number of jobs arriving from outside the network at queue j per unit time, if we model our network as a single traffic junction, all the vehicles will leave the network after being served at the junction. Such queuing networks are called *single-hop* queueing networks. However, we also wish to model networks where vehicles travel between different junctions. These are called multi-hop networks. To this end, $P = (P_{i,j} : i \in \mathcal{J}, j \in \mathcal{J} \cup \{\emptyset\})$ as given above are the expected routing probabilities. Here $P_{i,j}$ gives the probability that a vehicle departing queue i joins queue j (i.e., $P_{i,j} = \lim_{t \rightarrow \infty} \sum_{\tau=0}^{t-1} \mathbb{E}\{p_{i,j}(\tau)\}$). Also, we let \emptyset represent an exit state so $P_{i,\emptyset}$ gives the probability a job at queue i departs the network. Under the above $\lambda = (\lambda_j : j \in \mathcal{J})$ satisfying

$$\lambda_j = a_j + \sum_{i \in \mathcal{J}} \lambda_i P_{i,j} \quad (3)$$

gives the rate of arrival of vehicles to queue j (including arrivals from within the network). A special case of a multi-hop network is one with fixed routing. Here the probability $P_{i,j}$ is either zero or one.

We let $\mu_j, j \in \mathcal{J}$ be the nominal service rate of queue j . (It is average number of vehicle that can be processed per unit of time in our discrete time model under ideal conditions. The service rate received can be impacted by the state and features of the queueing network. We will later elaborate on the concept of service at a queue.) This represents the mean number of vehicles served at queue j when the queue length is sufficiently large. We let the vector $\rho = (\rho_j : j \in \mathcal{J})$ be the load on each queue. Specifically, we have that:

$$\rho_j := \frac{\lambda_j}{\mu_j},$$

for $j \in \mathcal{J}$. It is now reasonable to ask what is the maximum load that the queueing network can maintain. It is well-known in many settings that this is given by the condition that $\rho \in \mathcal{C}$.

The number of departures at each time depends on the service rule used and the state of jobs in the queue. We let $X_j(t) = (X_{j,k}(t) : k = 1, \dots, Q_j(t))$ be the features of each vehicle at queue j at time t . The features of a vehicle are a choice of the algorithm designer, but could include information such as the position in the queue, delay, speed and importantly if the vehicle is part of a platoon. Further discussion on these features is given in Section 2.2. We let $N_j(t)$ be the number of vehicles that will be served at junction j given that $s_j(t) = 1$. We define:

$$\mu_j(X_j(t)) := \mathbb{E}[N_j(t) | X_j(t)] \quad (4)$$

which gives the expected number of vehicles to depart queue j if served with a vehicle having features $X_j(t)$ are waiting. We assume that

$$\mu_j(X_j(t)) \rightarrow \mu_j \quad \text{as} \quad |X_j(t)| \rightarrow \infty \quad (5)$$

i.e. as the queue becomes congested the service rate converges to a fixed rate. Given $S_j(t)$ as defined above we have that

$$D_j(t) = \sum_{\tau=1}^t N_j(\tau) s_j(\tau). \quad (6)$$

We assume that the set of features at each queue $(X_j(t) : j \in \mathcal{J})$ evolves as a discrete-time Markov chain. Recall that a Markov chain is positive recurrent if it reaches a state a positive fraction of the time. In the context of a queueing network this is the state where all queues are empty and thus all demand has been served (Asmussen et al., 2003, Chapter 1). Given this, we say that a scheduling policy is stable if this Markov chain is positive recurrent, otherwise, we say that the scheduling policy is unstable (Tassiulas and Ephremides, 1990; Dai, 1995; Bramson et al., 2021; Mandjes et al., 2017).

2.2. Max weight with learned features of vehicles and platoons

Max Weight Policy: Let us consider a fundamental Max Weight policy utilizing queue length for intersection $n \in \mathcal{N}$, where the weight of a turning movement $i \in \mathcal{J}_n$ is calculated as follows:

$$w_i(t) = Q_i(t) \quad (7)$$

We note that Eq. (7) is different from queue-based Back Pressure in Varaiya (2013). In Varaiya (2013), the weight takes into account the queue length of downstream movements, i.e. $\sum_j p_{i,j}(t)Q_j(t)$. Hence, a queue-based Max Weight allocates service at a timestep by maximizing the following objective function:

$$\text{maximize } \sum_{i \in \mathcal{I}_n} \mu_i w_i s_i \quad \text{over } s \in \mathcal{S}_n. \quad (8)$$

The *Max Weight policy* chooses the schedule that solves the above optimization problem.

Vehicles Platooning: At the beginning of every timestep, the intersection controllers will (*virtually*) group vehicles into platoons based on their locations in each turning movement $i \in \mathcal{I}$. Note here that if a vehicle cannot go through the intersection in the current timestep, its platoon formation will be re-calculated in the next timestep. There are two conditions to decide whether a pair of consecutive vehicles belongs to the same platoon: firstly, they can cross the intersection within the same presumable green timestep; and secondly, their headway is less than a predefined value. The first condition means that the size of a platoon must not be larger than the maximum capacity μ_i . The total number of platoons for movement i is denoted by $L_i(t)$. Each platoon is described by its length $Q_{i,m}(t)$, $m = \{1, \dots, L_i(t)\}$ and its delay which is the sojourn time of Head-Of-Platoon (HOP) vehicle in the link $T_{i,m}(t)$, $m = \{1, \dots, L_i(t)\}$. Note that a vehicle could be a platoon of itself if it does not satisfy the above conditions. Characteristics of platoons can be shown by the following expressions:

$$Q_{i,m}(t) \leq \mu_i \quad , m = 1, \dots, L_i(t) \quad (9)$$

$$\sum_{m=1}^{L_i(t)} Q_{i,m}(t) = Q_i(t) \quad (10)$$

$$T_{i,m}(t) \geq T_{i,m+1}(t) \quad , m = \{1, \dots, L_i(t) - 1\} \quad (11)$$

Vehicles and platoons' features: We note that in the simple queue-based Max Weight described above, each vehicle counts equally to weight at a queue, specifically, the weight of queue i , is Q_i . We expect that there are more features that influence the rate of service beyond the queue length, such as the characteristics of the platoon that vehicles belong to (i.e. sojourn time since the platoon joins the link or platoon current velocity). Therefore, we seek to develop a system that can assess and optimize the importance of each of these features while maintaining good stability properties. Hence, rather than have each vehicle count 1, we allow vehicles to each have an influence between 0 and 1 on the total weight of the queue. More formally, we seek to use a continuous and increasing function $\sigma : \mathbb{R} \rightarrow [0, 1]$. Specifically, we utilize the logistic function in this paper:

$$\sigma(z) = \frac{e^z}{1 + e^z}, \quad z \in \mathbb{R} \quad (12)$$

With each vehicle we assume there are statistics, $\mathbf{x} = (1, x_1, \dots, x_d) \in \mathbb{R}^{d+1}$. These statistics determine the characteristics (or context) of vehicles. We assume that the components x_1, \dots, x_d determine other features and that these are bounded by a maximum value x_{\max} .

We apply a parameter $\theta \in \mathbb{R}^{d+1}$ to each of these statistics. In particular, if $\mathbf{x}^{(k)}$ is the statistic associated with the k th vehicle then we set

$$z^{(k)} := \theta_0 + \theta_1 x_1^{(k)} + \dots + \theta_d x_d^{(k)} = \theta^\top \mathbf{x}^{(k)} \quad (13)$$

and

$$\sigma^{(k)} = \sigma(z^{(k)}) \quad (14)$$

While our theoretical results and simulations focus on fixed turning probabilities and platooning, we note that features could include other aspects such as the time of day or the anticipated turning probability of the vehicle. In other words, this model has sufficient flexibility to account for features not typically addressed in a standard pressure-based traffic control model. In this paper, $\theta \in \mathbb{R}^{d+1}$ is optimized by applying cross entropy method, which is discussed in the next section.

Max Weight policy with learned features: We associate a parameter $\theta_{i,j} \in \mathbb{R}^{d+1}$ with each movement. We now sum the statistics associated with vehicles to give a weight for the queue at each junction. Specifically the weight of queue (i, j) is:

$$w_{(i)} = \sum_{m=1}^{P_i} \sum_{k=1}^{Q_{i,m}} \sigma(\theta_{i,j}^\top \mathbf{x}^{(k)}) \quad (15)$$

for $\sigma^{(k)}$ as defined above. The optimal control at intersection n , $S_n^*(t)$, follows the below equation:

$$S_n^*(t) = \arg \max_{S_n \in \mathcal{S}_n} \left\{ \sum_{i \in \mathcal{I}_n} \mu_i w_i(t) s_i(t) \right\} \quad (16)$$

2.3. Cross entropy method

The cross entropy method (CEM) is a generic optimization technique. It is a zeroth order method, i.e. we do not have gradients.¹ So, for instance, it works well on combinatorial optimization problems, as well as reinforcement learning. Its convergence properties have also been well studied in academic literature (Costa et al., 2007)

The principle: Assume that we want to maximize a function $S(x)$ over $x \in \mathcal{X}$. We then sample random variables X_1, \dots, X_N from \mathcal{X} according to some parameterized distribution $p(x|\theta)$. Assuming that we can fit the parameter θ given data $(X^{(1)}, \dots, X^{(M)})$. Let $\hat{\theta}$ be the fit to this data. Finally, $\rho \in (0, 1)$ is an *importance* parameter used by CEM. The basic CEM procedure is now as follows:

1. **Sample** X_1, \dots, X_N from $p(\cdot|\theta)$ and evaluate $S(X_1), \dots, S(X_N)$.
2. **Order** the sample $X^{(1)}, \dots, X^{(N)}$ so $S(X^{(1)}) \geq S(X^{(2)}) \geq \dots \geq S(X^{(N)})$.
3. **Fit** $\hat{\theta}$ to the top $M = \lfloor \rho N \rfloor$ samples, $X^{(1)}, \dots, X^{(M)}$ and set $\theta \leftarrow \hat{\theta}$.

The above three steps are repeated until convergence. The fit step of the CEM uses a specific optimization to find $\hat{\theta}$. In particular, $\hat{\theta}$ maximizes the cross entropy of the sample $X^{(1)}, \dots, X^{(M)}$.² But essentially $\hat{\theta}$ is just the maximum likelihood estimator of the data $X^{(1)}, \dots, X^{(M)}$ (assuming it was IID from distribution $p(\cdot|\theta)$ for some θ).

The update for θ can be smoothed out. Here we replace the update with

$$\theta \leftarrow (1 - \alpha)\theta + \alpha\hat{\theta}$$

for some $\alpha \in (0, 1]$.

Proposed CEM-based Max Weight learning algorithm: In Section 2.2, we note that each vehicle in each movement has a range of features $x = (1, x_1, \dots, x_d) \in \mathbb{R}^{d+1}$. In this paper, we pick three features:

- Sojourn time since vehicle's HOP join the link i of movement (i, j) .
- Binary feature indicates whether the vehicle's HOP is stopping and waiting at the intersection or not.
- Binary feature indicates whether the vehicle's platoon can pass the intersection in the next time step or not, assuming that the corresponding signal phase turns green. We can easily estimate this feature considering the current position and speed of the vehicle, as well as the maximum capacity $\mu_{i,j}$.

While the first two features are motivated by Wu et al. (2017) and Liu and Gayah (2022), the third feature follows our observation. We noted that platoons that can potentially pass the intersection in the next time-step need to play a more significant role compared to those that cannot. Consequently, for each intersection, CEM needs to optimize the set of $\theta_{(i,j) \in \mathcal{J}} \in \mathbb{R}^4$. Our pseudo-code of CEM is presented in Algorithm 1.

Algorithm 1: Cross Entropy Method

Initializing:

- $M_0 = [0 \quad 0 \quad 0 \quad 0]$, $\Sigma_0 = I_4$, $N = 16$, $\rho = 0.25$,
 $\alpha = 0.75$
- Iteration $i = 0$

Sample: Generate $\theta_1, \dots, \theta_N$ with mean M_i and covariance Σ_i

While $\text{Var}[D(\theta^{(1)}), \dots, D(\theta^{(N)})] \geq 1(s/\text{veh})$, **do:**

- Simulate the proposed Max Weight algorithm for each of the parameter set $\theta_n, n = 1, \dots, N$
 - Let the delay $D_n, n = 1, \dots, N$ is the vehicle's average delay of the n^{th} sample
 - Order the sample $\theta^{(1)}, \dots, \theta^{(N)}$ so $D(\theta^{(1)}) \leq D(\theta^{(2)}) \leq \dots \leq D(\theta^{(N)})$
 - Let $\theta' = \theta^{(1)}, \dots, \theta^{(\rho N)}$, $M' = \text{Mean}(\theta')$, $\Sigma' = \text{Cov}(\theta')$
 - $M_{i+1} = (1 - \alpha)M_i + \alpha M'$, and
 $\Sigma_{i+1} = (1 - \alpha)\Sigma_i + \alpha \Sigma'$
 - $i = i + 1$
-

3. Theoretical results

While we note that methods such as the Cross Entropy Method do not, in general, have established theoretical convergence results, it is still possible to argue that the class of proposed Max Weight policies have good stability behavior for a broad range of different queueing models as described in Section 2. This established that the policy class is sufficiently expressive.

The following theorem establishes the stability of the proposed Max Weight in single-hop networks.

¹ Caveat: though one may (or may not) need gradient methods in the sub-routine that maximizes the cross entropy.

² Note the one could instead use other alternatives to the cross entropy.

Theorem 1. For a single-hop network with load belonging to the capacity region, that is $\rho \in \mathcal{C}$, the proposed Max Weight algorithm is stable for all choices of θ with $\theta_1 > 0$.

The above result shows the proposed Max Weight is maximally stable for all single-hop networks. In the context of a road traffic scenario, this shows that the policy will achieve the best throughput possible at individual junctions.

When networks of junctions are connected together, Max Weight is not in general known to be stable under subcritical loads (although in practice it can often be an effective choice). For networks of queues, the Back Pressure algorithm is often considered. However, as we see here the proposed Max Weight policy can also be applied to obtain stability.

Theorem 2. For each multihop network with $\rho \in \mathcal{C}$, there exists a choice of θ such that the proposed Max Weight is stable.

Here we see that the proposed Max Weight policy can also stabilize the network if parameters are appropriately learned. We note that there are also features of the Max Weight algorithm that must be learned, specifically routing probabilities and service rates.

Next, we see that with an appropriate choice of features, the proposed Max Weight policy can express any queue-size-based policy for a bounded set of queue sizes.

Theorem 3. Any policy on bounded queue sizes can be expressed with the proposed Max Weight algorithm.

The above results show that the proposed Max Weight can be very expressive for low to moderate queue sizes. Thus, with an appropriate set of features, this class of policies can perform more specific and effective decision-making rules that may account for nuanced aspects of decision-making. For instance, in our simulations, we apply features such as the speed of vehicles or whether vehicles are part of a platoon approaching a junction.

3.1. Proof of theoretical results

In this section, we apply a fluid model analysis to prove [Theorems 1 and 2](#). This fluid model is a set of ordinary differential equations which we define in [Section 3.1.1](#). The theoretical justification for the fluid model, [Proposition 2](#), is given in [proven in Appendix A](#). The proofs of [Theorems 1 and 2](#) are then provided in [Appendices B and C](#), while we will prove [Theorem 3](#) in [Section 3.1.2](#).

3.1.1. Fluid model

The fluid model associated with the queueing model described in [Section 2](#) is given by a continuous-time processes $q_j(t), a_j(t), d_j(t)$ with $j \in \mathcal{J}$ and $t \in \mathbb{R}_+$ where, for each $j \in \mathcal{J}$, $a_j(t)$ and $d_j(t)$ are non-decreasing, Lipschitz continuous with $a_j(0) = d_j(0) = 0$ and $q_j(t)$ is positive and where the following conditions are satisfied:

$$q_j(t) = q_j(0) + a_j(t) - d_j(t) \quad (17)$$

$$a_j(t) = a_j t + \sum_{i \in \mathcal{J}} d_i(t) P_{i,j} \quad (18)$$

$$\left(d'_j(t) / \mu_j : j \in \mathcal{J} \right) \in \operatorname{argmax}_{\sigma \in S} \sum_{j \in \mathcal{J}} w_j q_j(t) \sigma_j \quad \text{over } \sigma \in S. \quad (19)$$

In [Proposition 2](#) which we state and prove in [Appendix A](#), we show that the above fluid model is the limiting model for our model in [Section 2](#). We notice that the above fluid model has the same behavior as a weighted form of the Max Weight algorithm. This means that for large queue sizes, the system behaves in a way that is comparable to standard Max Weight. The implication of this is that we can then invoke Max Weight arguments to prove stability in the fluid system. In particular, after establishing [Proposition 2](#), the proof of [Theorem 1](#) is a standard argument for Max Weight. The proof of fluid stability of Max Weight in multi-hop networks is less standard and can be found in [Theorem 2 of Bramson et al. \(2021\)](#).

Thus we must establish stability in our fluid model above in order to prove the stability of the corresponding queueing model. A fluid model is said to be *stable* if for all solutions there exists a time $t_0 > 0$ such that, for all initial states with $\sum_{j \in \mathcal{J}} q_j(0) \leq 1$, all solutions of the fluid model equations, [\(17\)–\(19\)](#), satisfy

$$\sum_{j \in \mathcal{J}} q_j(t) = 0, \quad \forall t \geq t_0.$$

It has been well-known since [\(Dai, 1995\)](#) (see also [Bramson \(2008\)](#)) that, under appropriate conditions, the stability of a fluid model will imply a positive recurrence of the associated queueing network. In our case, we employ the following proposition.

Proposition 1. Suppose that the fluid model [\(17\)–\(19\)](#) is stable. Then the corresponding queueing network model given in [\(2\)–\(6\)](#) is positive recurrent.

We refer the reader to [Bramson \(2008\)](#) for the proof of this proposition.

Table 2
Scenario 1 - Balanced traffic demand settings in low, medium, and high demand.

High/Medium/Low demands (veh/s)	North	East	South	West
North	–	0.167 0.136 0.104	0.278 0.226 0.174	0.111 0.09 0.069
East	0.111 0.09 0.069	–	0.167 0.135 0.104	0.278 0.226 0.174
South	0.278 0.226 0.174	0.111 0.09 0.069	–	0.167 0.136 0.104
West	0.167 0.136 0.104	0.278 0.226 0.174	0.111 0.09 0.069	–

3.1.2. Proof of Theorem 3

Proof. The proof of Theorem 3 is relatively straightforward. Suppose π is a policy defined as a function of queue sizes less than some constant q_{\max} . If the queue sizes are bounded there is only a finite set of queue sizes that can be achieved. Suppose we associate features with each such state. That is for each queue size state q we have a feature x_q where $x_q = 1$ if the queueing network is in state q and $x_q = 0$ otherwise. If queue j is served under policy π when in state q then we can set the parameter associated with x_q at queue j to be 1, i.e. $\theta_{x_q}^j = 1$, otherwise we set $\theta_{x_q}^j = 0$. Further suppose that the function $\sigma(z)$ is chosen such that $\sigma(1) = 1$ and $\sigma(0) = 0$. Notice that then under the proposed Max Weight policy in this case, when the queueing network is in state q only queues which are served under policy π will be served because these are the only queues that receive a positive weight. Thus we have encoded the decisions of π within this set of features. \square

Remark 1. Comparison with BackPressure. In the implementation of BackPressure large queue sizes must build up in order to reduce queue sizes.

4. Numerical experiments

In this section, we evaluate the performance of our proposed Max Weight platoon pressure-based signal control policy, called Learning Optimal Max Weight (LOMW), via SUMO simulation (Lopez et al., 2018) and compare it with existing pressure-based control policies in Wongpiromsarn et al. (2012) (the first Back Pressure based signal control algorithm), Wu et al. (2017), and Liu and Gayah (2022) (the latest pressure-based signal control algorithm). We call those compared policies Queue-based Back Pressure (QBP), Delay-based Max Weight (DMW) and Delay-based Back Pressure (DBP), respectively.

We conducted comprehensive simulation scenarios to demonstrate the effectiveness of our proposed algorithm. Our simulations are divided into three parts. Firstly, we provide results regarding CEM for several traffic demand settings. There are two variants of LOMW in this part: one uses the single parameter set for all intersections, other uses distinctive parameter sets for each intersection. We find that two LOMW variants perform equivalently and outperform other algorithms. From the insights from the first part, we then seek to find a single set of parameters that can fit every traffic demand and every intersection. Finally, we apply the parameter set found in the second part to the large-scale Melbourne CBD network. The timestep size is set to 20 s for all algorithms. Similar to Liu and Gayah (2022), we observe that the maximum service rate for both left turns and through/right turns is 1800 veh/h/lane. Finally, the vehicle length is 5 m and we utilize SUMO's default Krauss car following model (Krauß, 1998).

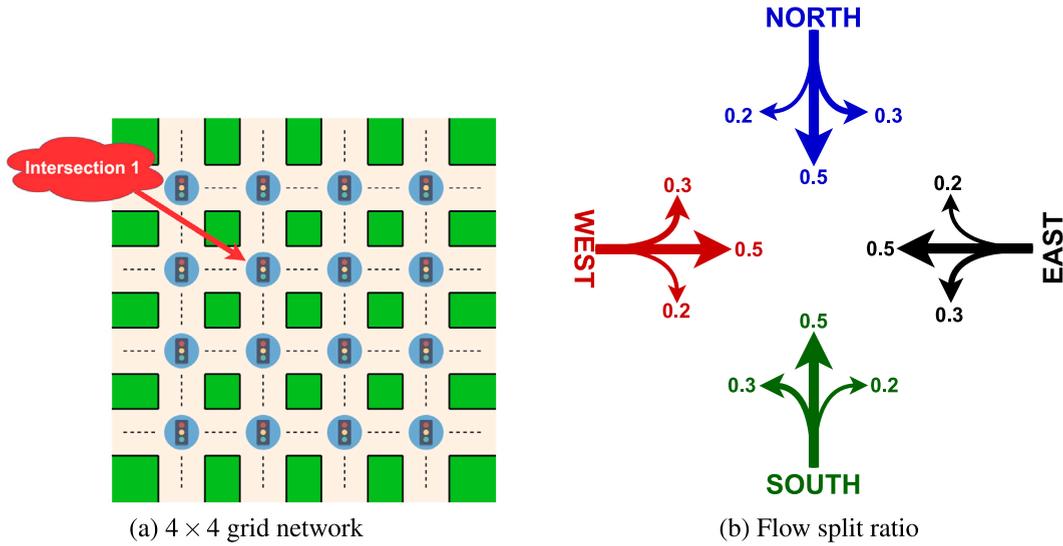
Finally, as mentioned in Section 2, we choose three features to obtain the weight of the proposed control scheme in this study:

- Sojourn time since vehicle's HOP join the link i of movement (i, j) .
- Binary feature indicates whether the vehicle's HOP is stopping and waiting at the intersection or not.
- Binary feature indicates whether the vehicle's platoon can pass the intersection in the next time step or not, assuming that the corresponding signal phase turns green. We can easily estimate this feature considering the current position and speed of the vehicle, as well as the maximum capacity $\mu_{i,j}$.

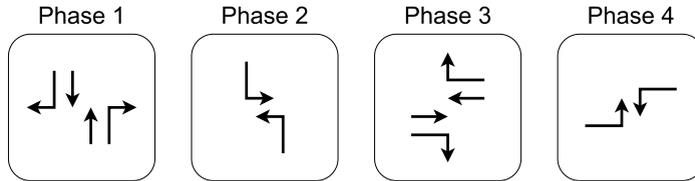
4.1. 4×4 grid network with separated CEM for each network scenario

As shown in Fig. 1, we utilize a grid 4×4 grid network with a total of 16 intersections. Each intersection has four phases as shown in Fig. 1c. Each link in the network consists of two lanes, and the rightmost lane is dedicated to right turns only. The speed limit is set to 13.89 m/s (50 km/h).

In this part, we set the two traffic demand scenarios: balanced traffic between East-West and North-South directions (scenario 1), and traffic demand originating from the North/South links higher than the East/West links (i.e. scenario 2 with imbalanced



Signal Configuration



(c) Signal Phase Configuration

Fig. 1. Network topology flow split setting and signal configuration.

Table 3
Scenario 2 - Imbalanced traffic demand settings in low, medium, and high demand.

High/Medium/Low demands (veh/s)	North	East	South	West
North	-	0.208 0.169 0.13	0.347 0.282 0.217	0.139 0.113 0.087
East	0.083 0.068 0.052	-	0.125 0.102 0.078	0.208 0.169 0.13
South	0.347 0.282 0.217	0.139 0.113 0.087	-	0.208 0.169 0.13
West	0.125 0.102 0.078	0.208 0.169 0.13	0.083 0.068 0.052	-

traffic demand). We also consider three traffic scenarios: light, medium and high traffic. Finally, as described in Fig. 1b, traffic flow originating from any direction will be split as follow: 50% of flow goes straight, 30% turns left, and 20% turns right. The detailed settings for all scenarios are presented in Tables 2 and 3. In those tables, the demands are shown in *veh/sec* unit and last for one hour.

Note that we describe the CEM learning algorithm for a single intersection in Algorithm 1. If we optimize the same set of $\theta_{(i,j)}$ for all intersections, then there will be only four parameters ($\theta_{(i,j) \in \mathcal{J}} \in \mathbb{R}^4$) and referred to as LOMW-4. For comparison, we also let each intersection have its own θ set. Hence, the set $\theta_{(i,j) \in \mathcal{J}} \in \mathbb{R}^{64}$ is optimized, and we refer to as LOMW-64.

Fig. 2 provides the convergence curves of CEM for all traffic scenarios. The existing algorithms serving as benchmarks are presented by horizontal lines in this figure. Each data point for LOMW-4 and LOMW-64 is the mean value of average vehicle delay over 72 simulations. The average vehicle delay is initially significant in all three traffic demand scenarios. It is because we initialize

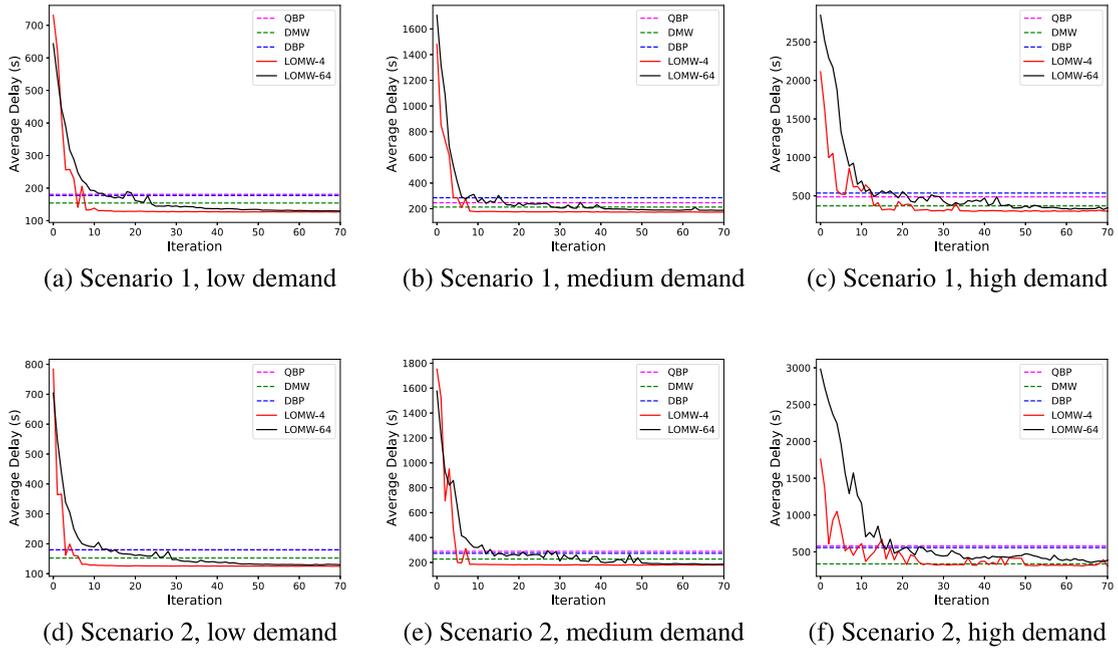


Fig. 2. Delay-CEM.

θ from a random mean and covariance value. However, the average vehicle delay converged quickly. Both LOMW-4 and LOMW-64 algorithms start to perform better than QBP, DBP and DMW after 25 iteration runs.

After finishing the cross entropy process, we pick the simulation results from the best set of parameters, then compare them with QBP and DBP. Note that we provide all the parameter sets resulting from the CEM in this numerical experiments section in Appendix D. Figs. 3 and 4 show the number of running vehicles in the network and vehicle mean speed during the simulation for all algorithms. It is easy to observe that for all cases, LOMW-4 and LOMW-64 outperform both QBP and DBP. Our proposed algorithms can deal with more vehicles in the network while keeping a better mean speed.

Note on computational complexity: in our CEM process, LOMW-64 requires more simulations per iteration compared to LOMW-4. Therefore LOMW-64 needs more computational power than LOMW-4. However, by running simulations in parallel, we can achieve a similar run time between CEM for LOMW-4 and LOMW-64. The CEM process in our experiment took place in Monash MASSIVE Super Cloud Computing.

4.2. 4 × 4 grid network aggregated CEM

Note that for all compared metrics, a significant finding is that LOMW-64 does not perform better than LOMW-4. It is a significant finding because we may only need a single set of $\theta_{(i,j)}$ for the whole network to obtain optimal results. In this section, we further exploit a single CEM procedure to find a single set of parameters that are suitable for all traffic scenarios. The modified CEM algorithm is then described in Algorithm 2.

Fig. 5 provides the average delay between all algorithms under different traffic experiments. For every single experiment, we observe that the aggregated LOMW-4 performs slightly worse than each separated LOMW-4 and LOMW-64. However, the aggregated LOMW-4 still outperforms QBP, DBP, and DMW. Therefore, this section provides that LOMW-4 is versatile and can perform well under different traffic scenarios by using a single set of parameters.

Moreover, Fig. 6 provides the average time-space density of upstream and downstream links at intersection 1. For QBP and DBP, there are several occasions when the density of the whole upstream links is high. In contrast, LOMW-4 and LOMW-64 can keep the low density for the whole simulation time.

Finally, we will test the effectiveness of aggregated LOMW-4 by applying the optimized parameters in the large-scale Melbourne CBD network in the next section (see Fig. 7).

4.3. Melbourne CBD network

As provided in Fig. 8, Melbourne CBD network consists of 38 intersections. All of the roads are bi-directional. Noticeably, King Street and Russell Street, as highlighted in blue color in Fig. 8, have three lanes in each direction. All other roads have two lanes in

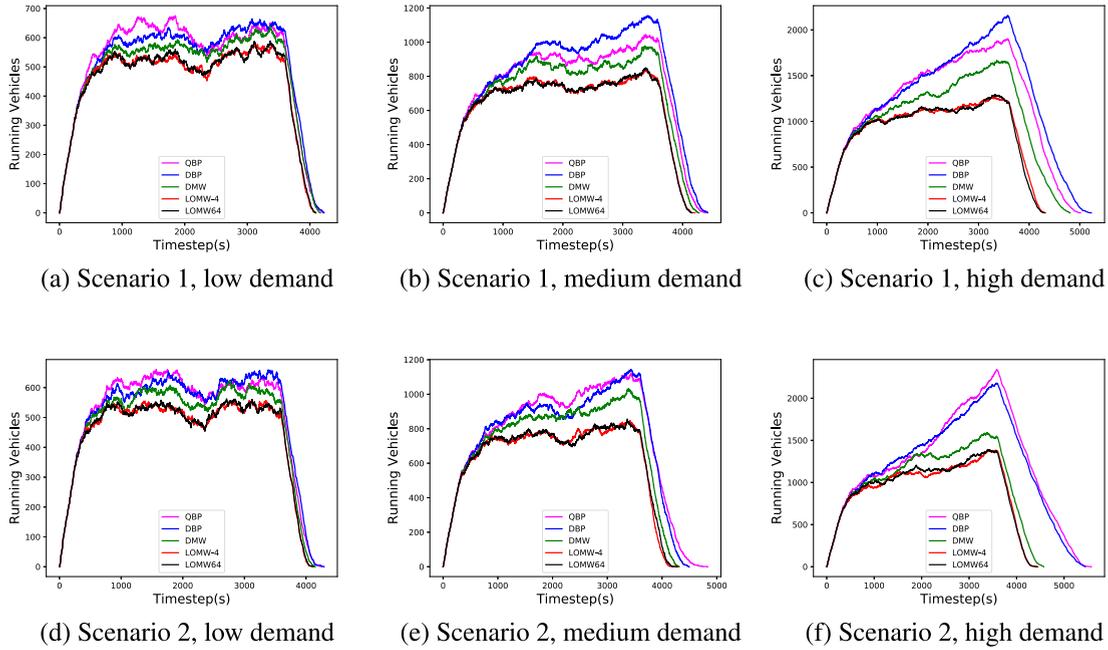


Fig. 3. Number of vehicles in the network for QBP, DBP, LOMW-4, LOMW-64.

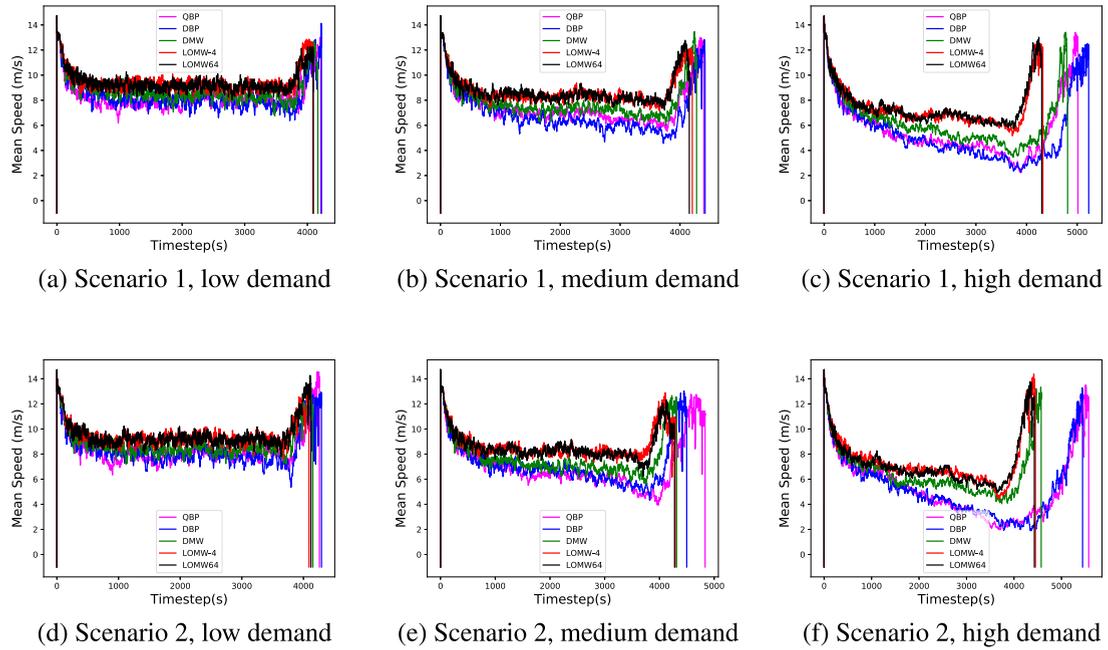


Fig. 4. Mean speed for QBP, DBP, LOMW-4, LOMW-64.

each direction. The signal phase configuration remains the same as the one provided in Fig. 1c. We simulate the peak-hour traffic for 4 h from 6AM to 10AM. There are a total of 70 400 trips in our simulation and the second hour is the one with most traffic. We generated the trips based on the OD demand data collected from Victoria’s SCATS dataset (DATAVIC, 2019). This dataset was thoroughly calibrated based on the work in Dantsuji et al. (2022).

Looking at the number of vehicles in the network (Fig. 9), in the first hour, when travel demand is moderate, all control schemes still perform well. However, when getting sudden high demands in the second hour, LOMW-4 remains the lowest number of vehicles compared to other algorithms. Noticeably, DMW cannot stabilize the traffic when it comes to the second hour. We observe that

Algorithm 2: Modified Cross Entropy Method for Aggregated LOMW**Initializing:**

- $M_0 = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}$, $\Sigma_0 = I_4$, $N = 16$, $\rho = 0.25$,
- $\alpha = 0.75$
- Iteration $i = 0$

Sample: Generate $\theta_1, \dots, \theta_N$ with mean M_i and covariance Σ_i

While $Var[D(\theta^{(1)}), \dots, D(\theta^{(N)})] \geq 1(s/veh)$, **do:**

- Simulate the proposed Max Weight algorithm **for each network traffic scenario** and for each of the parameter set $\theta_n, n = 1, \dots, N$
- Let the delay $D_n, n = 1, \dots, N$ is **the total vehicle's average delay of all scenarios** of the n^{th} sample
- Order the sample $\theta^{(1)}, \dots, \theta^{(N)}$ so $D(\theta^{(1)}) \leq D(\theta^{(2)}) \leq \dots \leq D(\theta^{(N)})$
- Let $\theta' = \theta^{(1)}, \dots, \theta^{(\rho N)}$, $M' = Mean(\theta')$, $\Sigma' = Cov(\theta')$
- $M_{i+1} = (1 - \alpha)M_i + \alpha M'$, and
 $\Sigma_{i+1} = (1 - \alpha)\Sigma_i + \alpha \Sigma'$
- $i = i + 1$

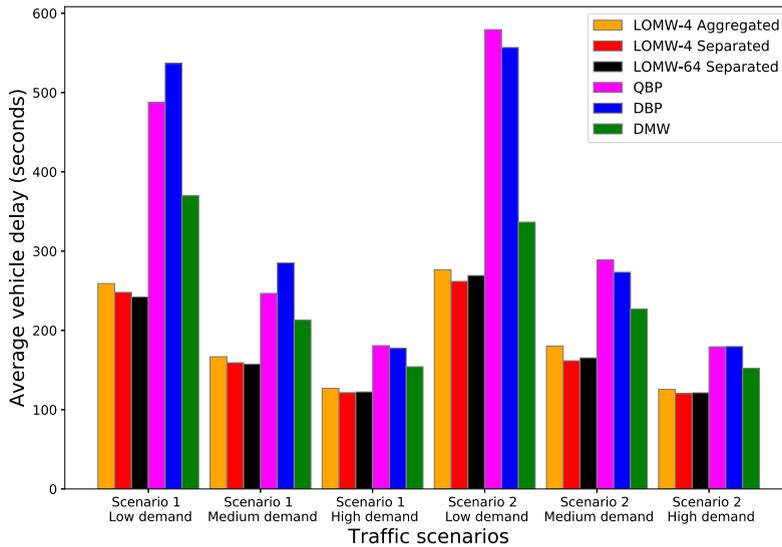


Fig. 5. Average vehicle delay (seconds) with different algorithms under different traffic scenarios.

the algorithm causes several gridlocks in the network, which surges the number of running vehicles. Consequently, LOMW-4 also outperforms other schemes in terms of vehicle delay, which is provided in Fig. 10.

5. Conclusion

This paper proposed a stable, optimal Max Weight platoon pressure-based network signal control algorithm under a connected vehicle environment. Compared to Back Pressure, the Max Weight-based signal control algorithm has advantages in terms of simplicity while still providing proven stability. Our proposed Max Weight signal control algorithm residing in the individual intersection controller leverages the V2X-based connectivity to collect information of the approaching vehicles such as their position and speed before (virtually) forming them into platoons for the purpose of managing traffic at intersections. The simulation results in SUMO showed the ability of the cross-entropy method to optimize and achieve a set of parameters for a large-scale network which is generic for all the intersections in that network and remains applicable across various traffic conditions. Our results showed that the proposed algorithm outperformed other well-known pressure-based algorithms which can be achieved with a single small set of parameters optimized by the proposed CEM learning algorithm. The results hold not only in the trained network environments but also in the large-scale Melbourne CBD test network. It is thus encouraging that learning is transferred between junctions of the network to improve performance. From a theoretical perspective, we provided proof of the stability of our proposed algorithm. In terms of practical application, it is evident that incorporating a small amount of learning into state-of-the-art algorithms can substantially improve network delays and overall performance.

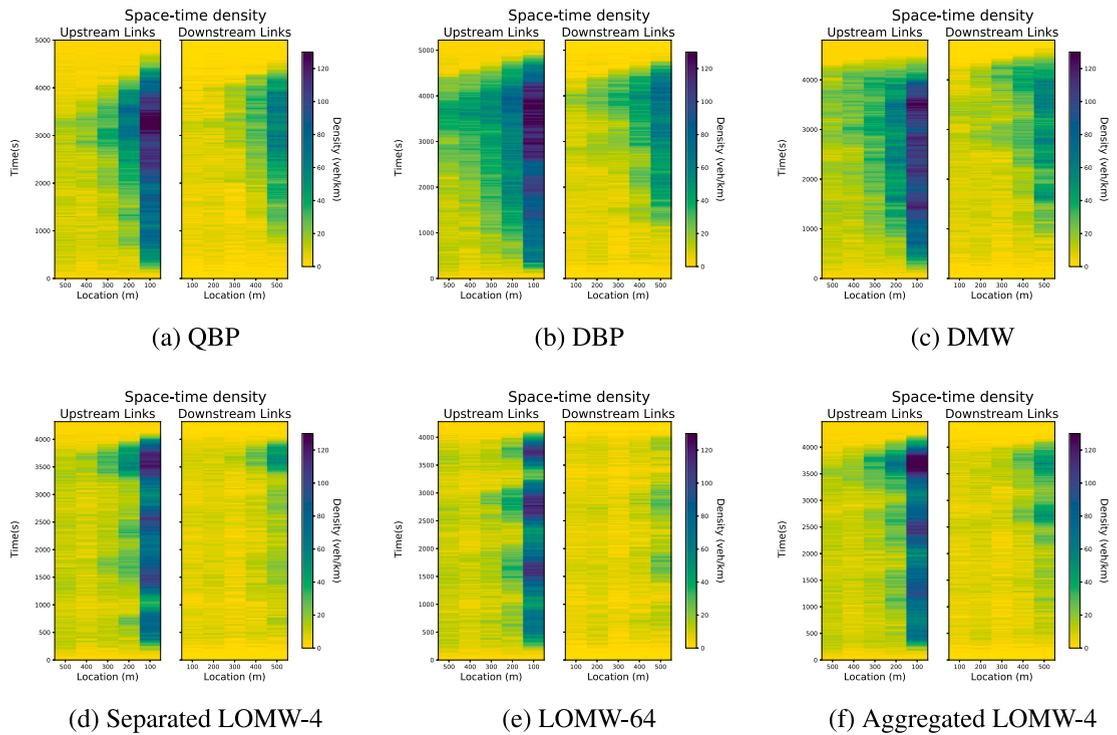


Fig. 6. Average time-space density at upstream and downstream links at intersection 1 under scenario 1 with high demand.

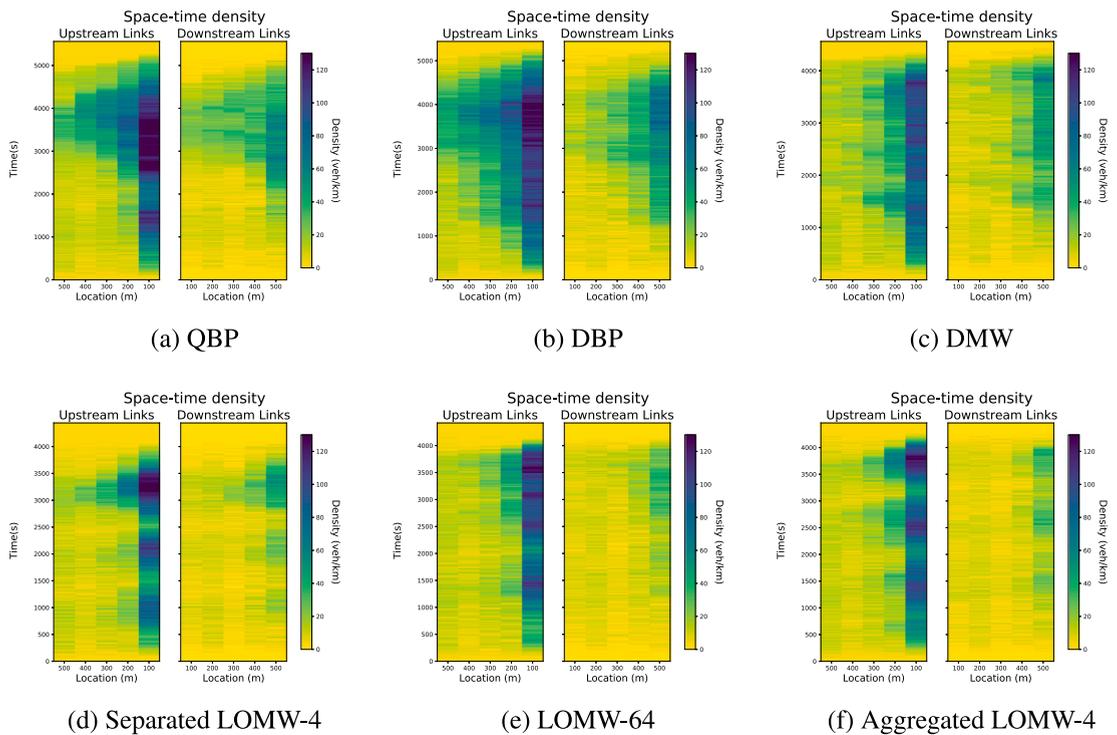


Fig. 7. Average time-space density at upstream and downstream links at intersection 1 under scenario 2 with high demand.

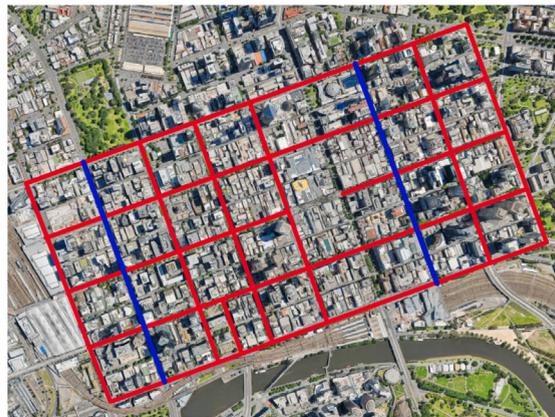


Fig. 8. Simulation network in Melbourne CBD.

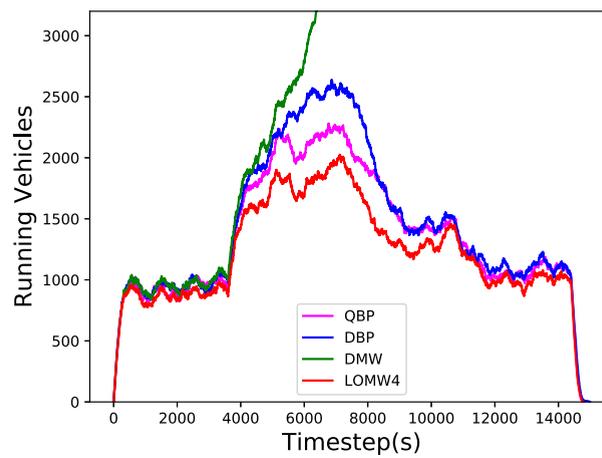


Fig. 9. Number of vehicles in the network.

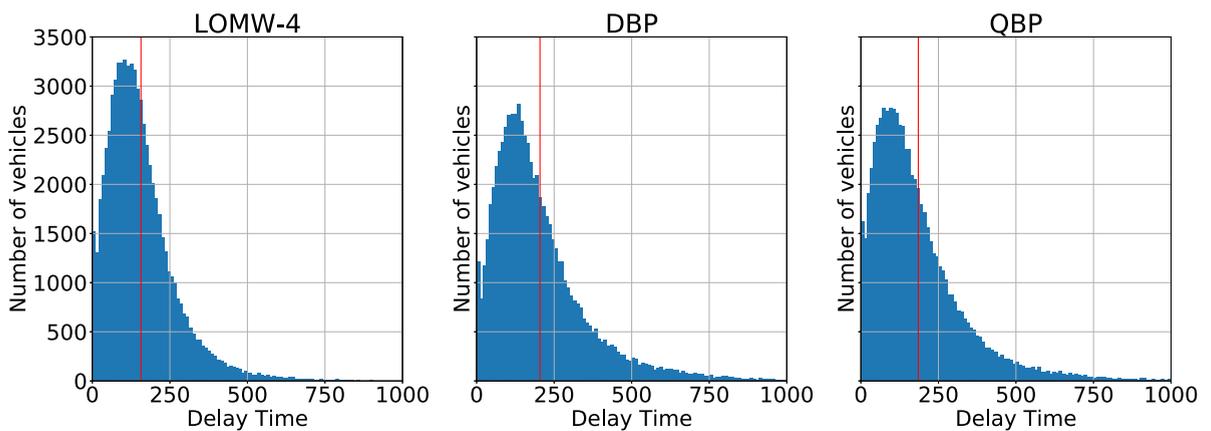


Fig. 10. Delay histogram (with the red vertical lines provide the average vehicle delay in the network).

Furthermore, mixed traffic with partial connectivity has been investigated by [Liu and Gayah \(2022\)](#), where the authors provided a comprehensive work to prove the effectiveness of a pressure-based algorithm under mixed traffic conditions with at least 30%

V2I connectivity. Their findings are also applicable to our proposed platoon-based scheme in a similar mixed traffic environment utilizing standard traffic state estimation methods as in Liu and Gayah (2022). Also note that in the case of partial V2V connectivity, the network will have more platoons with shorter lengths, which does not affect our method or change our conclusion.

There are several directions to improve the algorithms in future work. Firstly, this proposed algorithm can comprise several vehicle/platoon characteristics to calculate the movement pressure. Therefore, it is necessary to thoroughly study those characteristics in order to choose the significant ones. Secondly, the algorithm will be more practical if the output is the time of each phase within a cycle (Le et al., 2015). Finally, another possible direction is to design a framework that integrates the perimeter control at the network's boundary and the proposed algorithm for the internal signals.

CRedit authorship contribution statement

The Anh Hoang: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Neil Walton:** Writing – review & editing, Methodology, Formal analysis, Conceptualization. **Hai L. Vu:** Writing – review & editing, Writing – original draft, Validation, Supervision, Methodology, Data curation, Conceptualization.

Acknowledgments

This work was supported by the Australian Research Council (ARC) Discovery Grant DP190102134 and Industrial Transformation Research Hub IH180100010 (SPARC Hub). The financial and in-kind support of Austroads and Monash University, Australia are also gratefully acknowledged.

Appendix A. Convergence to fluid models

We derive the fluid model equations, given in Section 3.1.1, from scaled limits of the queueing network equations of switched queueing networks, operating under the Max Weight with Featurized Jobs. Our approach follows the standard fluid limit approach in Bramson (2008). A similar fluid limit proof for Max Weight is given in Bramson et al. (2021) and we adapt elements of that proof to the case of proposed Max Weight.

We consider a Max Weight queueing network with features as described in Section 2 with c jobs in the system initially. That is $c = |\mathcal{Q}^c(0)| = \sum_{j \in \mathcal{J}} \mathcal{Q}_j^c(0)$. (Here note that we apply a superscript to $\mathcal{Q} = \mathcal{Q}^c$ not make the dependence on c explicit.) We employ notation $A_j^c, D_j^c, E_j^c, Q_j^c, S_j^c, \Phi_{j,j'}^c$, and Π_j^c , for $j, j' \in \mathcal{J}$, analogous to that introduced in Section 2. We place these processes on the same probability space so that, regardless of the initial queue size c , they have the same external arrival, job size, and routing processes:

$$E_j^c(t) = E_j(t), \quad S_j^c(t) = S_j(t), \quad \text{and} \quad \Phi_{j,j'}^c(t) = \Phi_{j,j'}(t), \quad j, j' \in \mathcal{J}, \quad c \in \mathbb{N}.$$

For $j, j' \in \mathcal{J}$, we introduce the scaled processes

$$a_j^c(t) = A_j^c(ct)/c, \quad d_j^c(t) = D_j^c(ct)/c, \quad e_j^c(t) = E_j^c(ct)/c, \quad q_j^c(t) = Q_j^c(ct)/c, \quad (20a)$$

$$s_j^c(t) = S_j^c(ct)/c, \quad \phi_{j,j'}^c(t) = \Phi_{j,j'}^c(ct)/c, \quad (20b)$$

for $t \in \{0, c^{-1}, 2c^{-1}, 3c^{-1}, \dots\}$, and we interpolate linearly for other values of $t \in \mathbb{R}_+$.

As $c \rightarrow \infty$, these scaled processes converge to the fluid model Eqs. (17)–(19) in the following sense:

Proposition 2. *There exists a set G with $\mathbb{P}(G) = 1$ such that, for all $\omega \in G$, any scaled subsequence $(a_j^{c_i}, d_j^{c_i}, e_j^{c_i}, q_j^{c_i}, s_j^{c_i}, \phi_{j,j'}^{c_i}, \pi_j^{c_i} : j, j' \in \mathcal{J})$, $c_1 < c_2 < \dots$, of switched networks under the weighted Max Weight policy, contains a further subsequence that converges uniformly on compact time intervals. Moreover, any such limit $(a_j, d_j, e_j, q_j, s_j, \phi_{j,j'}, \pi_j : j, j' \in \mathcal{J})$ is a Lipschitz continuous process satisfying the weighted Max Weight fluid model Eqs. (17)–(19).*

Note that, since almost sure convergence implies convergence in distribution, Proposition 2 implies tightness/relative compactness and characterizes the weak convergent limits of the sequence $\{(a_j^c, d_j^c, e_j^c, q_j^c, s_j^c, \phi_{j,j'}^c, \pi_j^c : j, j' \in \mathcal{J})\}_{c \in \mathbb{N}}$. The proof of Proposition 2 is straightforward but somewhat technical. We provide a short proof.

Proof of Proposition 2. In order to demonstrate Proposition 2, we recall that, for each $j, j' \in \mathcal{J}$,

$$(E_j(t) - E_j(t-1) : t \in \mathbb{N}), \quad (S_j(t) - S_j(t-1) : t \in \mathbb{N}), \quad \text{and} \quad (\Phi_{j,j'}(t) - \Phi_{j,j'}(t-1) : t \in \mathbb{N})$$

are collections of i.i.d. random variables with respective means a_j, p_j^{-1} and $P_{j,j'}$. Therefore, by the (Functional) Strong Law of Large Numbers, on a set G_1 with $\mathbb{P}(G_1) = 1$,

$$e_j^c(t) \rightarrow a_j t, \quad s_j^c(t) \rightarrow p_j t, \quad \text{and} \quad \phi_{j,j'}^c(t) \rightarrow P_{j,j'} t \quad (21)$$

as $c \rightarrow \infty$, for $j, j' \in \mathcal{J}$, with convergence being uniform on compact time intervals.

By the Arzelà-Ascoli Theorem, any sequence of equicontinuous functions $\bar{X}^{c_i}(t)$ on $[0, T]$, $T \in (0, \infty)$, with $\sup_{c_i} |\bar{X}^{c_i}(0)| < \infty$, has a converging subsequence with respect to the uniform norm. We will show that the sequences $\{(a_j^c, d_j^c, e_j^c, q_j^c, s_j^c, \phi_{j,j'}^c : j, j' \in \mathcal{J})\}_{c \in \mathbb{N}}$ satisfy both conditions for any $\omega \in G_1$.

Since $|\bar{Q}^{c_i}(0)| = 1$ and the other functions are initially 0, the supremum is clearly bounded. In order to show equicontinuity, we first note that the sequences $e_j^c, s_j^c, \phi_{j,j'}^c$ converge uniformly on compact sets by the Functional Strong Law, and so these functions are equicontinuous.

To show equicontinuity for the remaining functions, note that D_j^c has bounded increments, and so their rescaled analogs are uniformly Lipschitz continuous; this implies equicontinuity of the corresponding sequences. Since q_j^c and a_j^c are the sum of a bounded number of such equicontinuous functions, they too are equicontinuous. Therefore, since the conditions of the Arzelà-Ascoli Theorem are met for $\omega \in G_1$, every subsequence $\{(a_j^{c_i}, d_j^{c_i}, e_j^{c_i}, q_j^{c_i}, s_j^{c_i}, \phi_{j,j'}^{c_i} : j, j' \in \mathcal{J})\}_c$ has a further subsequence that converges uniformly on $[0, T]$. Moreover, since these sequences of functions are uniformly Lipschitz, so are their limits.

We need to show that the fluid model equations in (17)–(19) are satisfied for all such limits. The equations in (17) follows directly from the queueing network equations in (2) and from (21). Similarly (18) follows directly from (4).

In order to show that (19) holds, take $\omega \in G_1$ and consider a sequence (d^{c_i}, q^{c_i}) that converges uniformly on compact sets to (d, q) . Notice that the proposed Max Weight objective converges uniformly to the weighted Max Weight objective (19). To see this notice that the weights in the proposed Max Weights are given by:

$$W_j^c(t) = w_j \sum_{k=1}^{Q_j^c(t)} \sigma(\theta_j^\top x^{c,(k)}).$$

For any queue such that $q_j^c(t) = Q_j^c(t)/c \rightarrow q_j(t)$ with $q_j(t)$ it must be that the first feature (which is the position of the vehicle in the queue) tends to infinity while all other features are assumed bounded. We know that for the sigmoid function $\sigma(z) \rightarrow 1$ as $z \rightarrow \infty$ and so we have that

$$\frac{W_j^c(t)}{c} = \frac{w_j}{c} \sum_{k=1}^{Q_j^c(t)} \sigma(\theta_j^\top x^{c,(k)}) \xrightarrow{c \rightarrow \infty} w_j q_j(t).$$

(Notice if $q_j(t) = 0$ the above limit is also zero.) From this, we see that in the limit the rate of service allocated to each junction is given by a weighted Max Weight allocation and so (19) holds. \square

Appendix B. Proof of Theorem 1

The proof of the stability of the fluid model follows a Lyapunov stability argument. Here we consider the Lyapunov function

$$L(q) = \frac{1}{2} \sum_{j \in \mathcal{J}} \frac{w_j}{\mu_j} q_j^2.$$

Also, we note that in the case of single-hop networks the fluid model (17)–(19) satisfies the following differential equation

$$\frac{dq_j}{dt} = a_j - d_j'(t),$$

for $j \in \mathcal{J}$.

Further we note that since $a \in \mathcal{C} = \langle \mathcal{S} \rangle^\circ$, there exists a positive constant ε such that

$$\begin{aligned} \varepsilon &= \max_{s \in \mathcal{S}} \min_{j \in \mathcal{J}} w_j (s_j - \rho_j) \\ &= \max_{s \in \mathcal{S}} \min_{q: \sum_j q_j = 1} \sum_{j \in \mathcal{J}} w_j q_j (s_j - \rho_j) \\ &\leq \min_{q: \sum_j q_j = 1} \max_{s \in \langle \mathcal{S} \rangle} \sum_{j \in \mathcal{J}} w_j q_j (s_j - \rho_j) \\ &= \min_{q: \sum_j q_j = 1} \max_{s \in \mathcal{S}} \sum_{j \in \mathcal{J}} w_j q_j s_j - \sum_{j \in \mathcal{J}} w_j q_j \rho_j. \end{aligned} \tag{22}$$

The inequality above uses the Minimax Theorem. Also in the final equality, we note that the maximum of a linear optimization over $\langle \mathcal{S} \rangle$ and \mathcal{S} achieve the same value since linear programs achieve their optimum on an extreme point.

Thus applying the chain rule:

$$\begin{aligned} \frac{dL}{dt} &= \sum_{j \in \mathcal{J}} \frac{\partial L}{\partial q_j} \frac{dq_j}{dt} \\ &= \sum_{j \in \mathcal{J}} \frac{w_j}{\mu_j} q_j(t) (a_j - d_j'(t)) \\ &= \sum_{j \in \mathcal{J}} w_j q_j(t) \left(\rho_j - \frac{d_j'(t)}{\mu_j} \right) \end{aligned}$$

$$\begin{aligned}
 &= \sum_{j \in \mathcal{J}} w_j q_j(t) \rho_j - \max_{s \in \mathcal{S}} \sum_{j \in \mathcal{J}} w_j q_j(t) s_j \\
 &\leq -\varepsilon \sum_{j \in \mathcal{J}} q_j(t) \\
 &\leq -\varepsilon \sqrt{\sum_{j \in \mathcal{J}} \frac{1}{w_j}} \sqrt{\sum_{j \in \mathcal{J}} w_j q_j(t)^2} \\
 &= -\varepsilon' L(q(t))^{\frac{1}{2}}.
 \end{aligned}$$

in the above expression, in the 4th equality, we apply the fluid model condition (19); we apply (22) in the first inequality, and then Cauchy–Schwartz in the second inequality. In the final equality we take $\varepsilon' = \varepsilon \sqrt{\sum_{j \in \mathcal{J}} 1/w_j}$ and apply the definition of $L(q)$. From the above expression, we can see that if $q(t) \neq 0$ then $L(q(t))$ is decreasing. This also implies that if $q(t_0) = 0$ for some t_0 then $q(t) = 0$ for all $t \geq t_0$.

Now integrating the above expression gives

$$2L(q(T))^{\frac{1}{2}} - 2L(q(0))^{\frac{1}{2}} = \int_0^T \frac{1}{L(q(t))^{\frac{1}{2}}} \frac{dL(q(t))}{dt} dt \leq - \int_0^T \varepsilon' dt = -\varepsilon' T.$$

Thus

$$L(q(T)) \leq \left(L(q(0))^{\frac{1}{2}} - \varepsilon' T \right)_+^2$$

from this we see that $q(t_0) = 0$ for some $t_0 = L(q(0))^{\frac{1}{2}}/\varepsilon'$ and thus $q(t) = 0$ for all $t \geq t_0$.

We have now proved that the fluid model is stable for all $\rho \in \mathcal{C}$. By Proposition 2, we know that the fluid model is the limit of the underlying queueing model, and thus by Proposition 1 the stability of the fluid model implies stability of the queueing model for $\rho \in \mathcal{C}$, as required. \square

Appendix C. Proof of Theorem 2

Proof. By Proposition 1 above we employ the fluid model equations. Specifically we analyze the fluid model equations (17)–(19). We focus on a specific parameter choice where $w_j = 1/\rho_j$ for each j . Thus the fluid model condition (19) becomes:

$$\left(d'_j(t)/\mu_j : j \in \mathcal{J} \right) \in \operatorname{argmax}_{\sigma \in S} \sum_{j \in \mathcal{J}} q_j(t) \frac{\sigma_j}{\rho_j} \text{ over } \sigma \in S.$$

For thus we define the following Lyapunov function

$$L(q(t)) := \max_{\sigma \in \mathcal{S}} \sum_{j \in \mathcal{J}} q_j(t) \left(\frac{\sigma_j}{\rho_j} - 1 \right) = \sum_{j \in \mathcal{J}} q_j(t) \left(\frac{d'_j(t)}{\lambda_j} - 1 \right).$$

The final equality holds by the fluid model condition above.

Observe that since $\rho \in \langle \mathcal{S} \rangle^\circ$ the above Lyapunov function is positive whenever $q(t) \neq 0$. We will show that this Lyapunov function converges to zero.

Now, notice that for $\delta < 0$ it holds that

$$\frac{L(q(t+\delta)) - L(q(t))}{\delta} \leq \sum_{j \in \mathcal{J}} \frac{q_j(t+\delta) - q_j(t)}{\delta} \left(\frac{d'_j(t)}{\lambda_j} - 1 \right)$$

This is hold because $(d'_j(t+\delta) : j \in \mathcal{J})$ is optimal for queue sizes $(q_j(t+\delta) : j \in \mathcal{J})$ whereas $(d'_j(t) : j \in \mathcal{J})$ is suboptimal. Letting $\delta \nearrow 0$, gives the first inequality below, which we then expand over a sequence of inequalities (each of which we discuss below):

$$\begin{aligned}
 \frac{dL}{dt} &\leq \sum_{j \in \mathcal{J}} q'_j(t) \left(\frac{d'_j(t)}{\lambda_j} - 1 \right) \\
 &= \sum_{j \in \mathcal{J}} \left(a_j + \sum_{j' \in \mathcal{J}} d'_{j'}(t) P_{j'j} - d'_j(t) \right) \left(\frac{d'_j(t)}{\lambda_j} - 1 \right) \\
 &= \sum_{j \in \mathcal{J}} a_j \frac{d'_j(t)}{\lambda_j} - \sum_{j \in \mathcal{J}} \lambda_j \left(\frac{d'_j(t)}{\lambda_j} \right)^2 + \sum_{j, j' \in \mathcal{J}} \lambda_j P_{j, j'} \left[\frac{d'_j(t)}{\lambda_j} \cdot \frac{d'_{j'}(t)}{\lambda_{j'}} \right] \\
 &\quad - \sum_{j \in \mathcal{J}} \left(a_j + \sum_{j' \in \mathcal{J}} d'_{j'}(t) P_{j'j} - d'_j(t) \right) \\
 &= \sum_{j \in \mathcal{J}} a_j \frac{d'_j(t)}{\lambda_j} - \sum_{j \in \mathcal{J}} \lambda_j \left(\frac{d'_j(t)}{\lambda_j} \right)^2
 \end{aligned}$$

$$\begin{aligned}
 & + \sum_{j,j' \in \mathcal{J}} \lambda_j P_{j,j'} \left[-\frac{1}{2} \left(\frac{d'_j(t)}{\lambda_j} - \frac{d'_{j'}(t)}{\lambda_{j'}} \right)^2 + \frac{1}{2} \left(\frac{d'_j(t)}{\lambda_j} \right)^2 + \frac{1}{2} \left(\frac{d'_{j'}(t)}{\lambda_{j'}} \right)^2 \right] \\
 & - \left(\sum_{j \in \mathcal{J}} a_j - \sum_{j \in \mathcal{J}} d'_j(t) P_{j\omega} \right) \\
 = & \sum_{j \in \mathcal{J}} a_j \frac{d'_j(t)}{\lambda_j} - \frac{1}{2} \sum_{j,j' \in \mathcal{J}} \lambda_j P_{j,j'} \left(\frac{d'_j(t)}{\lambda_j} - \frac{d'_{j'}(t)}{\lambda_{j'}} \right)^2 - \frac{1}{2} \sum_{j \in \mathcal{J}} a_j \left(\frac{d'_j(t)}{\lambda_j} \right)^2 \\
 & - \frac{1}{2} \sum_{j \in \mathcal{J}} \lambda_j P_{j\omega} \left(\frac{d'_j(t)}{\lambda_j} \right)^2 + \frac{1}{2} \sum_{j \in \mathcal{J}} \left\{ -\lambda_j + \sum_{j' \in \mathcal{J}} \lambda_{j'} P_{j',j} + a_j \right\} \left(\frac{d'_j(t)}{\lambda_j} \right)^2 \\
 & - \left(\frac{1}{2} \sum_{j \in \mathcal{J}} a_j + \frac{1}{2} \sum_{j \in \mathcal{J}} \lambda_j P_{j\omega} - \sum_{j \in \mathcal{J}} d'_j(t) P_{j\omega} \right) \\
 = & \sum_{j \in \mathcal{J}} a_j \frac{d'_j(t)}{\lambda_j} - \frac{1}{2} \sum_{j,j' \in \mathcal{J}} \lambda_j P_{j,j'} \left(\frac{d'_j(t)}{\lambda_j} - \frac{d'_{j'}(t)}{\lambda_{j'}} \right)^2 - \frac{1}{2} \sum_{j \in \mathcal{J}} a_j \left(\frac{d'_j(t)}{\lambda_j} \right)^2 \\
 & - \frac{1}{2} \sum_{j \in \mathcal{J}} \lambda_j P_{j\omega} \left(\frac{d'_j(t)}{\lambda_j} \right)^2 - \left(\frac{1}{2} \sum_{j \in \mathcal{J}} a_j + \frac{1}{2} \sum_{j \in \mathcal{J}} \lambda_j P_{j\omega} - \sum_{j \in \mathcal{J}} d'_j(t) P_{j\omega} \right) \\
 = & -\frac{1}{2} \sum_{j,j' \in \mathcal{J}} \lambda_j P_{j,j'} \left(\frac{d'_j(t)}{\lambda_j} - \frac{d'_{j'}(t)}{\lambda_{j'}} \right)^2 - \frac{1}{2} \sum_{j \in \mathcal{J}} a_j \left(\frac{d'_j(t)}{\lambda_j} - 1 \right)^2 \tag{23} \\
 & - \frac{1}{2} \sum_{j \in \mathcal{J}} \lambda_j P_{j\omega} \left(\frac{d'_j(t)}{\lambda_j} - 1 \right)^2. \tag{24}
 \end{aligned}$$

The first equality above applied the fluid model Eq. (17). The 2nd equality simply rearranges terms. The third equality applies the identity $x y = [x^2 + y^2 - (x - y)^2]/2$ for $x = d'_j(t)/\lambda_j$ and $y = d'_{j'}(t)/\lambda_{j'}$. For the 4th equality we rearrange terms, and add and subtract $\frac{1}{2} \sum_{j \in \mathcal{J}} a_j (d'_j(t)/\lambda_j)^2$, and we note that $\sum_{j \in \mathcal{J}} a_j + \sum_{j \in \mathcal{J}} \lambda_j P_{j\omega}$ follows from the traffic equations, (3). For the 5th equality, we observe that the term in curly brackets is zero by the traffic Eqs. (3). In the final equality, we notice that terms can be collected together as quadratics.

Since $(d_j(t)/\mu_j : j \in \mathcal{J})$ belongs to the boundary of $\langle \mathcal{S} \rangle$ and $\rho = (\lambda_j/\mu_j : j \in \mathcal{J})$ belongs to the interior of $\langle \mathcal{S} \rangle$. There exists a fixed ε (depending only on ρ and \mathcal{S}) and there exists j^* such that

$$\frac{d'_{j^*}(t)}{\lambda_{j^*}} - 1 > \varepsilon. \tag{25}$$

Now consider a path $j_1, \dots, j_L = j^*$, with $L \leq |\mathcal{J}|$ and satisfying $a_{j_1} > 0$ and $P_{j_\ell j_{\ell+1}} > 0$ for $\ell = 1, \dots, L - 1$. Consequently, by (25), it either holds that

$$\left| \frac{D'_{j_{\ell^*-1}}(t)}{\lambda_{j_{\ell^*-1}}} - \frac{D'_{j_{\ell^*}}(t)}{\lambda_{j_{\ell^*}}} \right| > \frac{\varepsilon}{|\mathcal{J}|} \text{ for some } \ell^* \in \{2, \dots, L\}, \text{ or } \left| \frac{D'_{j_1}(t)}{\lambda_{j_1}} - 1 \right| > \frac{\varepsilon}{|\mathcal{J}|}. \tag{26}$$

(Notice that if (26) did not hold then by applying the triangle inequality to (25), then we would see that (26) could not hold. Thus leading to a contradiction.)

Applying this to (24) gives that,

$$\frac{dL}{dt} \leq -c \frac{\varepsilon^2}{|\mathcal{J}|^2}. \tag{27}$$

all t satisfying $q(t) \neq 0$, where $c > 0$ is an appropriate constant.

Notice that, $L(q(0))$ is uniformly bounded for all $q(0)$ satisfying $\sum_j q_j(0) = 1$. For such $q(0)$, (27) implies a uniform bound t_0 exists with $L(q(t)) = 0$ for all $t \geq t_0$. Consequently, $q(t) = 0$ for all $t \geq t_0$. Thus the fluid model is stable. By Proposition 1, the proposed Max Weight queueing network is therefore positive recurrent. \square

Appendix D. LOMW-4 and LOMW-64 parameter sets resulting from the cross entropy method for balanced and imbalanced traffic demand, i.e. scenarios 1 and 2, respectively.

D.1. LOMW-4

D.1.1. Scenario 1

See Table 4.

Table 4
LOMW-4 Scenario 1.

High traffic demand	Medium traffic demand	Low traffic demand
[-2.743 0.806 0.092 0.012]	[-4.242 4.161 3.72 -4.752]	[-4.987 3.171 0.646 -2.616]

Table 5
LOMW-4 Scenario 2.

High traffic demand	Medium traffic demand	Low traffic demand
[-3.513 1.581 2.219 -1.711]	[-2.322 2.942 0.54 -3.659]	[-3.567 4.24 0.209 -6.064]

Table 6
LOMW-64 Scenario 1.

High traffic demand	Medium traffic demand	Low traffic demand
[-2.974 2.882 -1.839 -1.933]	[-0.502 1.324 -1.237 -3.015]	[-3.746 0.793 0.611 -0.690]
[-1.33 2.177 0.453 -0.046]	[-3.227 1.49 3.238 0.078]	[-0.874 1.029 0.079 -2.125]
[-4.29 1.126 -0.603 0.988]	[-1.59 2.13 -0.819 -1.692]	[-0.342 1.147 1.052 -2.920]
[-1.582 1.529 1.98 -2.81]	[-1.615 3.21 -1.489 -4.134]	[-0.527 1.319 1.310 -3.859]
[-2.475 0.815 -0.904 0.778]	[0.909 1.422 3.233 -4.063]	[-3.658 1.176 -0.032 0.073]
[2.282 1.24 2.773 -5.176]	[-1.049 0.479 -0.87 -1.853]	[-2.621 2.132 0.426 -2.666]
[-3.441 1.987 -1.413 -4.575]	[-2.232 0.992 1.519 -2.328]	[-5.216 2.465 -0.620 -0.649]
[0.236 1.276 3.011 -3.431]	[-2.328 2.592 -1.872 -3.455]	[-1.379 0.847 1.875 -1.352]
[-3.138 1.113 0.330 -2.144]	[-2.037 1.697 3.166 -2.193]	[-3.647 1.286 2.502 -0.839]
[-1.04 0.043 1.568 -0.282]	[0.045 0.977 0.779 -3.117]	[-2.054 1.669 2.214 -2.214]
[-4.48 0.683 2.75 -1.08]	[-1.372 1.818 0.702 -1.934]	[-2.766 1.934 -0.680 -0.887]
[-3.036 0.303 0.172 0.648]	[0.577 1.957 1.484 -4.746]	[-5.712 1.870 2.234 0.088]
[-1.238 2.536 -0.409 -2.184]	[-2.399 1.735 3.327 -1.97]	[0.58 1.467 -0.573 -4.262]
[3.684 2.508 3.981 -0.985]	[0.256 1.346 2.234 -3.001]	[-4.493 2.289 2.803 -1.037]
[-1.96 0.872 -1.644 -0.034]	[-4.281 0.894 1.648 -0.05]	[-1.135 2.030 -0.825 -4.249]
[-1.523 0.93 -1.166 -1.741]	[-1.445 0.301 0.714 -1.669]	[-1.955 2.602 0.274 -3.497]

D.1.2. Scenario 2

See Table 5.

D.1.3. Aggregated LOMW-4

$$[-3.686 \quad 1.703 \quad 2.151 \quad -1.463]$$

D.2. LOMW-64 (Each row represents the parameter set applied to the corresponding intersection)

D.2.1. Scenario 1

See Table 6.

D.2.2. Scenario 2

See Table 7.

Table 7
LOMW-64 Scenario 2.

High traffic demand				Medium traffic demand				Low traffic demand			
-4.245	1.591	1.201	-0.157	-3.152	0.481	4.783	0.273	-1.613	1.562	0.214	-2.327
-1.041	1.467	1.715	-5.094	-3.110	1.593	-1.288	-0.833	-3.962	1.785	2.331	-0.326
-3.329	2.240	-0.563	-0.609	-2.982	2.038	2.582	-3.480	-3.232	0.852	2.912	-1.313
1.413	1.618	2.286	1.817	-2.909	0.787	1.502	0.727	-0.625	1.022	0.946	-2.472
-2.964	0.781	0.026	0.881	-1.104	1.038	0.480	-1.143	-2.886	0.972	3.043	-0.268
-3.305	0.507	1.509	-1.571	-1.444	0.08	2.494	-1.287	-3.372	1.867	1.448	-1.096
-2.809	1.838	-1.071	-1.362	-1.563	1.876	1.731	-2.904	-3.192	2.131	-0.912	-1.352
-1.670	0.311	0.774	-0.330	-6.890	2.095	0.710	1.943	-3.415	2.365	0.274	-1.858
0.170	1.761	3.183	-3.08	-1.453	1.336	-1.859	-1.229	-2.902	2.006	0.932	-1.986
-4.783	1.991	0.844	-2.101	-4.733	1.827	0.001	-0.078	-1.104	1.823	1.057	-4.23
1.055	0.403	0.730	-3.848	0.312	1.226	-0.799	-3.756	-4.518	2.376	-0.443	-0.708
-3.767	0.585	0.373	1.219	-1.330	0.767	2.748	-2.612	1.699	1.941	3.159	-6.443
-2.624	0.810	1.463	-2.671	-2.162	0.711	0.162	-0.391	-1.309	0.956	0.668	-1.450
-2.791	0.921	3.872	-0.829	-1.765	1.122	2.049	-2.147	-5.329	2.001	0.486	0.241
-3.511	2.377	-4.813	-0.868	-1.574	1.464	3.041	-2.355	-1.427	3.322	-0.143	-5.251
-2.759	0.934	-0.090	-0.583	-4.598	0.760	2.816	0.473	-1.224	0.541	3.278	-2.661

References

- Al Islam, S.B., Hajbabaie, A., 2017. Distributed coordinated signal timing optimization in connected transportation networks. *Transp. Res. C* 80, 272–285.
- Asmussen, S., Asmussen, S., Asmussen, S., 2003. *Applied Probability and Queues*, vol. 2, Springer.
- Bramson, M., 2008. Stability of queueing networks. *Probab. Surv.* 5 (none), 169–345.
- Bramson, M., D'Auria, B., Walton, N., 2021. Stability and instability of the maxweight policy. *Math. Oper. Res.* 46 (4), 1611–1638.
- Costa, A., Jones, O.D., Kroese, D., 2007. Convergence properties of the cross-entropy method for discrete optimization. *Oper. Res. Lett.* 35 (5), 573–580.
- Dai, J.G., 1995. On positive harris recurrence of multiclass queueing networks: A unified approach via fluid limit models. *Ann. Appl. Probab.* 5 (1), 49–77.
- Dantsuji, T., Hoang, N.H., Zheng, N., Vu, H.L., 2022. A novel metamodel-based framework for large-scale dynamic origin-destination demand calibration. *Transp. Res. C* 136, 103545.
- DATAVIC, 2019. Traffic signale volume.
- Dixit, V., Nair, D.J., Chand, S., Levin, M.W., 2020. A simple crowdsourced delay-based traffic signal control. *PLoS One* 15 (4), e0230598.
- Emami, A., Sarvi, M., Bagloee, S.A., 2021. Network-wide traffic state estimation and rolling horizon-based signal control optimization in a connected vehicle environment. *IEEE Trans. Intell. Transp. Syst.*
- Gregoire, J., Frazzoli, E., de La Fortelle, A., Wongpiromsarn, T., 2014a. Back-pressure traffic signal control with unknown routing rates. *IFAC Proc. Vol.* 47 (3), 11332–11337.
- Gregoire, J., Qian, X., Frazzoli, E., De La Fortelle, A., Wongpiromsarn, T., 2014b. Capacity-aware backpressure traffic signal control. *IEEE Trans. Control Netw. Syst.* 2 (2), 164–173.
- He, Q., Head, K.L., Ding, J., 2011. PAMSCOD: platoon-based arterial multi-modal signal control with online data. *Procedia Soc. Behav. Sci.* 17, 462–489.
- Krauß, S., 1998. Microscopic modeling of traffic flow: Investigation of collision free vehicle dynamics.
- Le, T., Kovács, P., Walton, N., Vu, H.L., Andrew, L.L., Hoogendoorn, S.S., 2015. Decentralized signal control for urban road networks. *Transp. Res. C* 58, 431–450.
- Le, T., Vu, H.L., Walton, N., Hoogendoorn, S.P., Kovács, P., Quejia, R.N., 2017. Utility optimization framework for a distributed traffic control of urban road networks. *Transp. Res. B* 105, 539–558.
- Li, L., Jabari, S.E., 2019. Position weighted backpressure intersection control for urban networks. *Transp. Res. B* 128, 435–461.
- Liang, X.J., Guler, S.I., Gayah, V.V., 2020. An equitable traffic signal control scheme at isolated signalized intersections using connected vehicle technology. *Transp. Res. C* 110, 81–97.
- Liang, X., Guler, S.I., Gayah, V.V., 2021. Decentralized arterial traffic signal optimization with connected vehicle information. *J. Intell. Transp. Syst.* 1–16.
- Lioris, J., Kurzhanskiy, A., Varaiya, P., 2016. Adaptive max pressure control of network of signalized intersections. *IFAC-PapersOnLine* 49 (22), 19–24.
- Lioris, J., Pedarsani, R., Tascikaraoglu, F.Y., Varaiya, P., 2017. Platoons of connected vehicles can double throughput in urban roads. *Transp. Res. C* 77, 292–305.
- Liu, H., Gayah, V.V., 2022. A novel max pressure algorithm based on traffic delay. *Transp. Res. C* 143, 103803.
- Lopez, P.A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., ner, E.W., 2018. Microscopic traffic simulation using SUMO. In: *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE.
- Mandjes, M., Patch, B., Walton, N., 2017. Detecting Markov chain instability: a Monte Carlo approach. *Stoch. Syst.* 7 (2), 289–314.
- McKeown, N., Mekkittikul, A., Anantharam, V., Walrand, J., 1999. Achieving 100% throughput in an input-queued switch. *IEEE Trans. Commun.* 47 (8), 1260–1267.
- Mercader, P., Uwayid, W., Haddad, J., 2020. Max-pressure traffic controller based on travel times: An experimental analysis. *Transp. Res. C* 110, 275–290.
- Nguyen, C.H., Hoang, N.H., Lee, S., Vu, H.L., 2021. A system optimal speed advisory framework for a network of connected and autonomous vehicles. *IEEE Trans. Intell. Transp. Syst.*
- Priemer, C., Friedrich, B., 2009. A decentralized adaptive traffic signal control using V2I communication data. In: *2009 12th International IEEE Conference on Intelligent Transportation Systems*. IEEE, pp. 1–6.
- Pumir, T., Anderson, L., Triantafyllos, D., Bayen, A.M., 2015. Stability of modified max pressure controller with application to signalized traffic networks. In: *2015 American Control Conference*. ACC, IEEE, pp. 1879–1886.
- Stebbins, S., Hickman, M., Kim, J., Vu, H.L., 2017. Characterising green light optimal speed advisory trajectories for platoon-based optimisation. *Transp. Res. C* 82, 43–62.
- Tassiulas, L., Ephremides, A., 1990. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. In: *29th IEEE Conference on Decision and Control*. IEEE, pp. 2130–2132.
- Ubierno, G.A., Jin, W.-L., 2016. Mobility and environment improvement of signalized networks through vehicle-to-infrastructure (V2I) communications. *Transp. Res. C* 68, 70–82.
- Varaiya, P., 2013. Max pressure control of a network of signalized intersections. *Transp. Res. C* 36, 177–195.
- Wang, P., Jiang, Y., Xiao, L., Zhao, Y., Li, Y., 2020. A joint control model for connected vehicle platoon and arterial signal coordination. *J. Intell. Transp. Syst.* 24 (1), 81–92.

- Wang, Q., Yuan, Y., Yang, X.T., Huang, Z., 2021. Adaptive and multi-path progression signal control under connected vehicle environment. *Transp. Res. C* 124, 102965.
- Wongpiromsarn, T., Uthaicharoenpong, T., Wang, Y., Frazzoli, E., Wang, D., 2012. Distributed traffic signal control for maximum network throughput. In: 2012 15th International IEEE Conference on Intelligent Transportation Systems. IEEE, pp. 588–595.
- Wu, J., Ghosal, D., Zhang, M., Chuah, C.-N., 2017. Delay-based traffic signal control for throughput optimality and fairness at an isolated intersection. *IEEE Trans. Veh. Technol.* 67 (2), 896–909.
- Zaidi, A.A., Kulcsár, B., Wymeersch, H., 2016. Back-pressure traffic signal control with fixed and adaptive routing for urban vehicular networks. *IEEE Trans. Intell. Transp. Syst.* 17 (8), 2134–2143.