PressLight: Learning Max Pressure Control to Coordinate Traffic Signals in Arterial Network

Hua Wei[†], Chacha Chen[‡], Guanjie Zheng[†], Kan Wu[†], Vikash Gayah[†], Kai Xu[§], Zhenhui Li[†]

[†]Pennsylvania State University, [‡]Shanghai Jiao Tong Univeriisity, [§]Shanghai Tianrang Intelligent Technology Co., Ltd
[†]{hzw77, gjz5038, jessieli}@ist.psu.edu, [†] kxw5389@psu.edu, [†] gayah@engr.psu.edu, [‡]chacha1997@sjtu.edu.cn,
[§]kai.xu@tianrang-inc.com

ABSTRACT

Traffic signal control is essential for transportation efficiency in road networks. It has been a challenging problem because of the complexity in traffic dynamics. Conventional transportation research suffers from the incompetency to adapt to dynamic traffic situations. Recent studies propose to use reinforcement learning (RL) to search for more efficient traffic signal plans. However, most existing RL-based studies design the key elements - reward and state - in a heuristic way. This results in highly sensitive performances and a long learning process.

To avoid the heuristic design of RL elements, we propose to connect RL with recent studies in transportation research. Our method is inspired by the state-of-the-art method max pressure (MP) in the transportation field. The reward design of our method is well supported by the theory in MP, which can be proved to be maximizing the throughput of the traffic network, i.e., minimizing the overall network travel time. We also show that our concise state representation can fully support the optimization of the proposed reward function. Through comprehensive experiments, we demonstrate that our method outperforms both conventional transportation approaches and existing learning-based methods.

CCS CONCEPTS

• **Computing methodologies** → **Artificial intelligence**; Control methods; • **Applied computing** → *Transportation.*

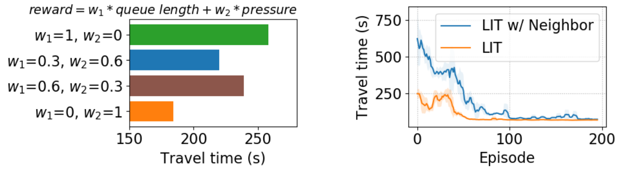
KEYWORDS

Deep reinforcement learning, traffic signal control, multi-agent system

ACM Reference Format:
Hua Wei[†], Chacha Chen[‡], Guanjie Zheng[†], Kan Wu[†], Vikash Gayah[†], Kai Xu[§], Zhenhui Li[†]. 2019. PressLight: Learning Max Pressure Control to Coordinate Traffic Signals in Arterial Network. In *The 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD ́19), August 4–8, 2019, Anchorage, AK, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3292500.3330949

KDD ’19, August 4–8, 2019, Anchorage, AK, USA

ACM ISBN 978-1-4503-6201-6/19/08…$15.00
https://doi.org/10.1145/3292500.3330949



(a) Performance w.r.t. reward (b) Convergence w.r.t. state

Figure 1: Performance of RL approaches is sensitive to reward and state. (a) A heuristic parameter tuning of reward function could result in different performances. (b) The method with a more complicated state (LIT [34] w/ neighbor) has a longer learning time but does not necessarily converge to a better result.

1 INTRODUCTION

Traffic signals coordinate the traffic movements at the intersection and a smart traffic signal control algorithm is the key to transportation efficiency. Traffic signal control remains an active research topic because of the high complexity of the problem. The traffic situations are highly dynamic, thus require traffic signal plans to be able to adjust to different situations.

Recently, people start to investigate reinforcement learning (RL) techniques for traffic signal control. Several studies have shown the superior performance of RL techniques over traditional transportation approaches [1, 2, 4, 25, 31, 32]. The biggest advantage of RL is that it directly learns how to take the next actions by observing the feedback from the environment after previous actions.

One major issue of current RL-based traffic signal control approaches is that the setting is often heuristic and lacks proper theoretical justification from transportation literature. This often results in highly sensitive performance w.r.t. the setting and leads to a long learning process. We elaborate on this issue by examining two fundamental elements in RL setting: reward and state.

First, various reward designs have been proposed in the literature. The reason is that travel time, the ultimate objective, is hard to optimize directly. Travel time is a long-term reward depending on a sequence of actions, thus the effect of one action can hardly be reflected in terms of travel time. People thus choose short-term rewards like queue length or delay to approximate the travel time [30]. So the reward function is often defined as a weighted sum of these terms [6, 9, 10, 25, 31]. However, as shown in Figure 1(a), tuning the weights on these terms could lead to largely different results in terms of travel time. Some literature [35] discusses how to define the reward by connecting with the existing transportation method,

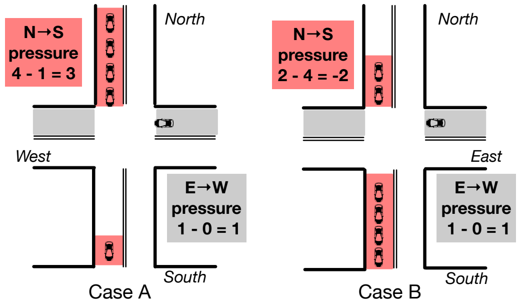


Figure 2: Illustration of max pressure control in two cases. In Case A, green signal is set in the North→South direction; in Case B, green signal is set in the East→West direction.

but they only focus on controlling a single intersection. In this paper, we focus on the multi-intersection control scenario.

Second, existing RL methods have a trend of using more complicated state representation. Recent studies use visual images to describe the full traffic situation at the intersection [25, 31], which results in the dimension of the state in the scale of thousands. In the single intersection scenario, [35] reveals that additional information is not always helpful. Similar conclusions can also be found in the multi-intersection scenario. As shown in Figure 1(b), complicated state definitions increase the learning time and may not necessarily bring significant gain. Note that we are not claiming that additional information is always not helpful. The choice of the state depends on the reward setting. Based on the reward design of LIT [35], neighboring information is not necessary in the case we show in Figure 1(b). The question is, could we justify theoretically how much information is enough in state definition in order to optimize the reward function?

The challenges we face in RL motivate us to look for support from transportation. In transportation literature, max pressure (MP) control is one of the state-of-the-arts in traffic signal control [16, 26]. The key idea of MP is to minimize the "pressure" of an intersection, which can be loosely defined as the number of vehicles on incoming lanes minus the number of vehicles on outgoing lanes. Figure 2 illustrates the concept of pressure. By setting the objective as minimizing the pressure of intersections, MP is proved to maximize the throughput of the whole road network[1]. However, the solution of MP is greedy, which leads to locally optimal solutions.

Our proposed solution is based on RL but theoretically grounded by MP method. The connection between RL and MP is that both approaches can essentially be framed as an optimization problem. In RL, long term reward is the objective for optimization and the solution is derived from trial-and-error search. In MP, the objective is to minimize pressure and the solution is derived from a greedy algorithm. Intuitively, if we set our reward function the same as the objective of MP, we can achieve the same result as MP. We first prove that under the assumption of no physical queue expansion, both our method and MP are maximizing throughput of the network. We further show that our method can relax the assumption on queue expansion and the conclusion still holds.

To further address the challenge on state design, we describe the system dynamics using the state features based on MP. MP provides evolution equations to formulate the state transition of the traffic as a Markov chain [28]. In RL, the Markov decision process formally describes the dynamics of an environment. By including the variables from the evolution equation into state definition in RL, the state is a sufficient statistic for the system dynamics.

We conduct comprehensive experiments using both synthetic data and real data. We test our method in different traffic flow and network structure scenarios. We demonstrate the power of RL methods over traditional transportation approaches as RL optimizes the objective through trial and error. Our method also consistently outperforms state-of-the-art RL methods, which shows that theoretically supported reward design is necessary and the concise state design leads to an efficient learning process. We further discuss several interesting policies learned by our method to show that our method can achieve coordination along arterial.

2 RELATED WORK

Individual Traffic Signal Control. Individual traffic signal control has been investigated extensively in the field of transportation, which tries to optimize the travel time or delay of vehicles [5, 11, 12, 15, 23], assuming that vehicles are arriving and moving in a specific pattern. Recently, reinforcement learning based methods attempt to address this problem by directly learning from the data [18, 32]. Earlier work using tabular Q-learning [3, 9] can only deal with discrete state representations. Recent work using deep RL [7, 14, 20, 25, 31, 34] can cope with more complex continuous state representation. [35] noticed that it is not always true that the more complex the state definitions are, the better the performance will be. In [35], they also investigated the proper reward design grounded by the individual intersection control method in transportation field. In this paper, we are focusing on the multi-intersection scenario.

Conventional Multi-intersection Traffic Signal Control. In conventional multi-intersection control, coordination can be achieved by setting a fixed offset (i.e., the time interval between the beginnings of green lights) among all intersections along an arterial [24]. In fact, it is not an easy task, given traffic of opposite directions usually cannot be facilitated simultaneously. To solve this problem, some optimization-based methods [17, 21] are developed to minimize vehicle travel time and/or the number of stops at multiple intersections. Instead of optimizing offsets, max pressure [26, 28] aims to maximize the throughput of the network so as to minimizing the travel time. However, these approaches still rely on assumptions to simplify the traffic condition and do not guarantee optimal results in the real world.

RL-based Multi-intersection Traffic Signal Control. Since recent advances in RL improve the performance on isolated traffic signal control [31, 35], efforts have been made to design strategies that control multiple intersections. *One way* is to consider jointly modeling the action between learning agents with centralized optimization [13, 25]. Since these methods [13, 25] need to negotiate between the agents in the whole network, they are computationally expensive. *Another way* is to use decentralized RL agents to control the traffic signals in the multi-intersection system [4, 8, 10]. Since each agent makes its own decision based on the information from

[1] Maximizing throughput equals to minimizing travel time under certain conditions and minimizing travel time is the final goal for most traffic signal control problems.

itself and neighboring intersections without centralized decision, decentralized methods may be more scalable and practicable. By plugging new intersection controllers into the system, the decentralized systems are easy to scale. Our proposed method also follows this direction.

We notice the recent trend to vary the definition of state and reward in RL for traffic signal control. Readers interested in the detailed comparison of the state and reward definitions can refer to [30]. We are the first RL method that is theoretically grounded by traditional transportation methods to coordinate the traffic signals along an arterial.

3 PRELIMINARIES

Definition 3.1 (Incoming lane and outgoing lane of an intersection). An incoming lane for an intersection is a lane where the traffic enters the intersection. An outgoing lane for an intersection is a lane where the traffic leaves the intersection. We denote the set of incoming lanes and outgoing lanes of an intersection as L_{in} and L_{out} respectively.

Definition 3.2 (Traffic movement). A traffic movement is defined as the traffic traveling across an intersection from one incoming lane to an outgoing lane. We denote a traffic movement from lane l to lane m as (l, m).

Definition 3.3 (Movement signal and phase). A movement signal is defined on the traffic movement, with green signal indicating the corresponding movement is allowed and red signal indicating the movement is prohibited. We denote a movement signal as $a(l, m)$, where $a(l, m) = 1$ indicates the green light is on for movement (l, m), and $a(l, m) = 0$ indicates the red light is on for movement (l, m). A phase is a combination of movement signals. We denote a phase as $p = \{(l, m)|a(l, m) = 1\}$, where $l \in L_{in}$ and $m \in L_{out}$.

In Figure 3, there are twelve incoming lanes and twelve outgoing lanes in the intersection. Eight movement signals (red and green dots around the intersection) comprise *four* phases to control the traffic movements for the intersection: *WE-Straight* (Going Straight from West and East), *SN-Straight* (Going Straight from South and North), *WE-Left* (Turning Left from West and East), *SN-Left* (Turning Left from South and North). Specifically, *WE-Left* allows two traffic movements. When phase #2 is activated, the traffic from l_E and l_W is allowed to turn left to corresponding outgoing lanes.

Definition 3.4 (Pressure of movement, pressure of intersection). The pressure of a movement is defined as the difference of vehicle density between the incoming lane and the outgoing lane. The vehicle density of a lane is defined as $x(l)/x_{max}(l)$, where $x(l)$ is the number of vehicles on lane l, $x_{max}(l)$ is the maximum permissible vehicle number on l. We denote the pressure of movement (l, m) as

$$w(l, m) = \frac{x(l)}{x_{max}(l)} - \frac{x(m)}{x_{max}(m)} \tag{1}$$

If all the lanes have the same maximum capacity x_{max}, then $w(l, m)$ is simply indicating *the difference between the incoming and outgoing number of vehicles.*

The pressure of an intersection i is defined as the sum of the absolute pressures over all traffic movements, denoted as:

$$P_i = |\sum_{(l,m)\in i} w(l, m)| \tag{2}$$

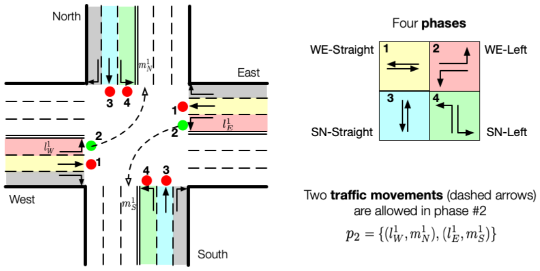

North
East
West
South
m_N^1
l_E^1
l_W^1
m_S^1
1
2
3
4
Four **phases**
WE-Straight
WE-Left
SN-Straight
SN-Left
Two **traffic movements** (dashed arrows) are allowed in phase #2
$p_2 = \{(l_W^1, m_N^1), (l_E^1, m_S^1)\}$

Figure 3: Phase and traffic movements in traffic signal control problem. Phase #2 is set in the example.

In Figure 2, the pressure of the intersection in Case A is $|3+1| = 4$, whereas the pressure of intersection in Case B is $| - 2 + 1| = 1$. In general, the pressure P_i indicates the degree of disequilibrium between the incoming and outgoing vehicle density. The larger P_i is, the more unbalanced the distribution of vehicles is.

Problem 1 (Multi-intersection traffic signal control)**.** In our problem, each intersection is controlled by an RL agent. At each time step t, agent i observes from the environment as its state o_i^t. Given the vehicle distribution and current traffic signal phase, the goal of the agent is to give the optimal action a (i.e., which phase to set), so that the reward r (i.e., the average travel time of all vehicles) can be maximized.

4 METHOD

4.1 Agent Design

First, we introduce the state, action and reward design for an agent that controls an intersection.

- **State (Observation)**. Our state is defined for one intersection, which equals to the definition of observation in multi-agent RL. It includes the current phase p, the number of vehicles on each outgoing lane $x(m)$ ($m \in L_{out}$), and the number of vehicles on each segment of every incoming lane $x(l)_k$ ($l \in L_{in}, k = 1 \ldots K$). In this paper, each lane is evenly divided into 3 segments ($K = 3$), and we denote the segment on lane l nearest to the intersection as the first segment $x(l)_1$.
- **Action**. At time t, each agent chooses a phase p as its action a_t from action set $\boldsymbol{A}$, indicating the traffic signal should be set to phase p. In this paper, each agent has four permissible actions, correspondingly four phases in Figure 3. Each action candidate a_i is represented as a one-hot vector. Note that in the real world the signal phases may organize in a cyclic way, while our action makes the traffic signal plan more flexible. Also, there may be different number of phases in the real world and four phases is not a must.
- **Reward**. We define the reward r_i as

$$r_i = -P_i, \tag{3}$$

where P_i is the ***pressure*** of intersection i, as defined in Equation (2).

Intuitively, the pressure P_i indicates the degree of disequilibrium between vehicle density on the incoming and outgoing lanes. By minimizing P_i, the vehicles within the system can be evenly distributed. Then the green light is effectively utilized so that the throughput is optimized.

4.2 Learning Process

In this paper, we adopt Deep Q-Network (DQN) as function approximator to estimate the Q-value function. To stabilize the training process, we maintain an experience replay memory as described in [19] by adding the new data samples in and removing the old samples occasionally. Periodically, the agent will take samples from the memory and use them to update the network.

5 JUSTIFICATION OF RL AGENT

To theoretically support the efficacy of our proposed method, we justify our reward and state design by showing that, in a simplified transportation system, the states we use can fully describe the system dynamics, and using Equation (3) as reward function in RL is equivalent to optimizing travel time as in the transportation methods. Some important notation is summarized in Table 1.

Table 1: Summary of notation.

Notation	Meaning
L_{in}	set of incoming lanes for an intersection
L_{out}	set of outgoing lanes for an intersection
(l, m)	a traffic movement from lane l to m
$x(l, m)$	number of vehicles leaving l and entering m
$x(l)$	number of vehicles on lane l
$x(l)_k$	number of vehicles on k-th segment of l
$x_{max}(m)$	maximum permissible vehicle number on lane m
$r(l, m)$	turning ratio of traffic movements from l to m
$c(l, m)$	discharging rate of movement (l, m)
$a(l, m)$	1 if the green light is on for movement (l, m), 0 otherwise

5.1 Justification for State Design

5.1.1 General description of traffic movement process as a Markov chain. Consider the arterial scenario described in Example 5.1.

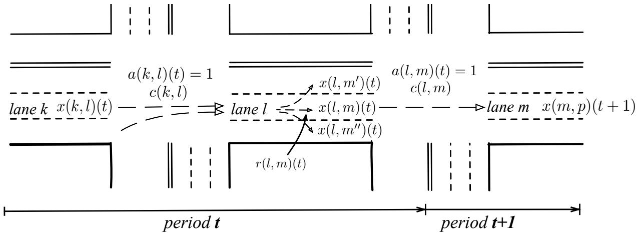


Figure 4: The transition of traffic movements.

Example 5.1. Figure 4 associates a distinct traffic movement with each incoming lane $l \in L_{in}$ and each $m \in Out_l$, where Out_l is the set of lanes output from lane l. Follow the notation from [28], let $x(l, m)(t)$ be the associated number of vehicles at beginning of period t, $X(t) = \{x(l, m)(t)\}$ is the *state* of the movement network, which we regard as states o^t in accordance with Section 4.1. There are two variables which are considered independent of $X(t)$:

- Turning ratio $r(l, m)$: $r(l, m)$ is an i.i.d. random variable indicating the proportion of vehicles entering m from l to the total vehicles on l.
- Discharging rate $c(l, m)$: For each (l, m), the queue discharging rate $c(l, m)$ is a non-negative, bounded, i.i.d. random variable, i.e., $c(l, m) \leq C(l, m)$, where $C(l, m)$ is the saturation flow rate.

At the end of each period t, an action $A^t = \{(l, m)|a^t(l, m)\}$ must be selected from the action set $\boldsymbol{A}^t$ as a function of X^t for use in period $(t + 1)$, indicating the agent will give green light for movements from l to m, see the bottom of Figure 4.

The evolution equations of $X(t)$ are developed in [26]. For each (l, m) and t, the evolution of $x(l, m)$ consists of receiving and discharging, and is captured by the following equation:

$$\begin{aligned} & x(l,m)(t+1) \\ &= x(l,m)(t) + \underbrace{\Sigma_{k \in In_l} min[c(k,l) \cdot a(k,l)(t),\ x(k,l)(t)] \cdot r(l,m)}_{receiving\ vehicles} \\ &- \underbrace{min\{c(l,m) \cdot a(l,m)(t),\ x(l,m)(t)\} \cdot \mathbf{1}(x(m) \leq x_{max}(m))}_{discharging\ vehicles}, \end{aligned} \tag{4}$$

where In_l represents the set of lanes input to l. For the second term in Equation (4), when l is the receiving lane, up to $x(k, l)$ vehicles will move from k if $a(k, l)(t) = 1$ and they will join (l, m) if $r(l, m) = 1$ For the third term in Equation (4), when traffic movement (l, m) is actuated, i.e., $a(l, m)(t) = 1$, up to $x(l, m)$ vehicles will leave l and be routed to m if there is no blockage on lane m, i.e., $x(m) \leq x_{max}(m)$, where $x_{max}(m)$ is the maximum permissible vehicle number on lane m.

Suppose the initial state $X(1) = x(l, m)(1)$ is a bounded random variable. Since $A(t) = a(l, m)(t)$ is a function of the current state $X(t)$, and $c(l, m)$ and $r(l, m)$ are all independent of $X(1), ..., X(t)$, the process $X(t)$ is a ***Markov chain***. The transition probabilities of the chain depend on the control policy.

5.1.2 Specification with proposed state definition. We can modify the traffic movement equation from lane-level to segment-level. We denote $x(l)_1$ as the number of vehicles on the segment l_1 closest to the intersection and $x(l)_2$ as the number of vehicles on the second closest segment, which is connected with l_1. Assume the vehicles change lanes for routing by the time it enters the lane l, i.e., $x(l, m) = x(l)$, and all vehicles on l_{i+1} enter next segment l_i during time t, then the movement process on the segment closest to the intersection can be written as:

$$\begin{aligned} x(l)_1(t+1) = &\ x(l)_1(t) + \ x(l)_2(t) \\ - &\ min\{c(l,m) \cdot a(l,m)(t),\ x(l)_1(t)\} \cdot \mathbf{1}(x(m) \leq x_{max}(m)). \end{aligned} \tag{5}$$

Equations for other segments can be derived in a similar way.

With the lane and segment movement evolution equations described above, the evolution of an individual intersection could be obtained, which is a combination of the equations of all the lanes involved. For a single intersection i, $c(l, m)$ is a constant physical feature of each movement, whereas $x(l)_1$, $x(l)_2$, and $x(m)$ are provided to the RL agent in our state definition. Hence, our state definition can fully describe the dynamics of the system.

5.2 Justification for Reward Design

5.2.1 Stabilization on traffic movements with proposed reward. Inspired by [26], we first relax its assumption on physical queue expansion in the arterial. Then the goal of our RL agents is proven to stabilize the queue length, thus maximizes the system throughput and minimizes the travel time of vehicles.

Definition 5.2 (Movement process stability). The movement process $X(t) = \{x(l,m)(t)\}$ is stable in the mean (and u is a stabilizing control policy) if for some $M < \infty$, the following holds:

$$\sum_{t=1}^{T} \sum_{(l,m)} E[x(l,m)(t)] < M, \quad \forall T \tag{6}$$

where E denotes expectation. Movement stability in the mean implies that the chain is positive recurrent and has a unique steady-state probability distribution for all T.

Definition 5.3 (Max-pressure control policy [26]). At each period t, the agent selects the action with maximum pressure at every state X: $\tilde{A}^*(X) = \arg\max_{\tilde{A}\in\boldsymbol{A}} \theta(\tilde{A}, X)$, where the pressure of $\tilde{A}$ is defined as

$$\theta(\tilde{A}, X) = \sum_{(l,m):a(l,m)=1} \tilde{w}(l,m),$$

and $\tilde{w}(l,m) = x(l) - x(m)$ is the pressure of each movement. In this paper, we use the tilde symbol for max-pressure policy, i.e., $\tilde{A}$, in order to differentiate it from a RL policy.

Theorem 5.4. *Without considering the physical queue expansion*[2]*, action $\tilde{A}^*$ selected by max-pressure control policy and action A^* selected by our RL policy are both stabilizing the system, whenever the average demand is admissible*[3].

Proof. For max-pressure control policy, Theorem 1 in [26] shows that given a time period $t = 1, \ldots, T$ there exists $m < \infty$ and $\epsilon > 0$ such that under $\tilde{A}^*$: $\epsilon \cdot \frac{1}{T}\sum_{t=1}^{T} E[X(t)] \le m + \frac{1}{T} \cdot E[X(1)]^2$, where $X(1)$ denotes the state when $t = 1$.

For an optimal RL control policy, the agent selects the action A with optimal $Q(A, X)$ at every state X:

$$A^*(X) = \arg\max_{A\in\boldsymbol{A}} Q(A, X). \tag{7}$$

where $Q_t(A, X) = E[r_{t+1} + \gamma r_{t+2} + \ldots |A, X]$ denotes the maximum total reward at state X by taking A at time t(in Equation (7), we neglect time t for simplicity). The difference between the pressure definition in RL reward and max-pressure is that our RL agent uses the weighted pressure considering maximum permissible vehicle number x_{max} in Equation (1). If we assume the lanes are in the same lenth $x_{max}(l)$, the stability result still holds for the normalized $x(l)$. □

Theorem 5.5. *Considering the physical queue expansion in the arterial environment, action A^* selected by our RL policy is also stabilizing the movement.*

Different from [26], we now establish the proof of Theorem 5.5, which removes the assumption of no physical queue expansion in the arterial environment. In the arterial environment:

- The maximum permissible vehicle number x_{max} on side street lane m^{side} is assumed to be infinite, hence the second term in Equation (1) is zero. Thus we have $w(l, m^{side}) = \frac{x(l)}{x_{max}(l)} > 0$.
- When the outgoing lane m^{main} along the arterial is saturated, the second term in Equation (1) is approximately 1 because of the queue expansion. Thus $w(l, m^{main}) \approx \frac{x(l)}{x_{max}(l)} - 1 < 0$.

This means when we consider the physical queue expansion in the arterial, $w(l, m^{side}) > w(l, m^{main})$, the control policy will restrict the queue spillback since it prohibits more vehicles to rush into the downstream intersection and block the movements of vehicles in other phases. Accordingly, M in Equation (6) can now be set to $M \le \sum_{t=1}^{T} \sum_{(l,m)} x_{max}(m)$.

5.2.2 Connection to throughput maximization and travel time minimization. Given that the traffic movement process of each intersection is stable, the system is accordingly stable. In an arterial environment without U-turn, vehicles that move from lane m to l would not move from l to m again, i.e., between $x(m, l)$ and $x(l, m)$ only one of them can exist under arterial network. Then the actions that RL agents take will not form gridlock or block the network, thus can efficiently utilize the green time. Within the given time period T, our RL agent can provide the maximum throughput, thus minimize the travel time of all vehicles within the system.

6 EXPERIMENT

We conduct experiments on CityFlow[4], an open-source traffic simulator that supports large-scale traffic signal control [33]. After the traffic data being fed into the simulator, a vehicle moves towards its destination according to the setting of the environment. The simulator provides the state to the signal control method and executes the traffic signal actions from the control method.[5]

6.1 Dataset Description

Both synthetic and real-world traffic flow data are used in our experiments. In a traffic dataset, each vehicle is described as (o, t, d), where o is origin location, t is time, and d is destination location. Locations o and d are both locations on the road network. Traffic data is taken as input for the simulator. All the data contains bi-directional and dynamic flows with turning traffic.

- **Synthetic data**. Four different configurations are tested as detailed in Table 2. This data is synthesized from a statistical analysis of real-world traffic patterns in Jinan and Hangzhou.
- **Real-world data**. We collect six representative traffic flow data from three cities to evaluate the performance of our model: Beaver Avenue in State College, USA; Qingdao Road in Jinan, China; four avenues in Manhattan, New York City, USA. Figure 5 shows the aerial view on these arterials. Detailed statistics of these datasets are listed in Table 3.

6.2 Experimental Settings

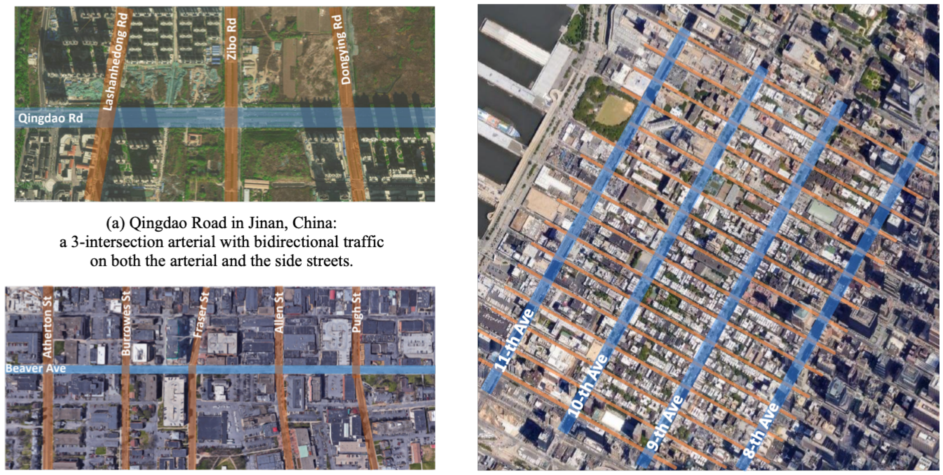
6.2.1 Environmental settings. Different road networks are configured. Besides a six-intersection arterial on which we primarily

[2] "Without physical queue expansion" means the vehicles will be considered to have no physical length in a queue.

[3] Intuitively, an admissible demand means the traffic demand can be accommodated by traffic signal control strategies, not including situations like long-lasting over-saturated traffic that requires perimeter control to stop traffic from getting in the system.

[4] http://cityflow-project.github.io

[5] Codes, public datasets and their preprocessing and statistical details can be found at: https://github.com/wingsweihua/presslight.
More datasets can be found at: http://traffic-signal-control.github.io



(a) Qingdao Road in Jinan, China: a 3-intersection arterial with bidirectional traffic on both the arterial and the side streets.

(b) Beaver Avenue in State College, Pennsylvania, USA: a 5-intersection arterial with unidirectional traffic on the arterial and bidirectional traffic on the side streets.

(c) 8-th, 9-th, 10-th and 11-th Avenue in New York City, USA: four 16-intersection arterials with uni-directional traffic on both the arterial and the side streets.

Figure 5: Real-world arterial network for the experiment.

experiment, arterials with larger scale and heterogeneous intersections (in Figure 7) are also tested.

The free-flow speed on the road segments is set to 40 kilometers/hour. Vehicles can always turn right when there is no conflicting traffic. Every time the phase switches, a 5-second combined yellow and all-red time are followed to clear the intersection.

Table 2: Configurations for synthetic traffic data

Config	Demand pattern	Arrival rate (vehicles/h/road)	Volume
1. Light-Flat	Flat	Arterial : 600 Side-street: 180	(Light)
2. Light-Peak	Peak		
3. Heavy-Flat	Flat	Arterial: 1400 Side-street : 420	(Heavy)
4. Heavy-Peak	Peak		

6.2.2 Evaluation metric. Following existing studies [31], we use the average **travel time** in seconds to evaluate the performance. The average travel time of all vehicles is the most frequently used measure in the transportation field [22], which is calculated as the average travel time of all vehicles spent in the system.

6.2.3 Compared methods. We compare our model with the following two categories of methods: transportation methods and RL methods. Note that all methods are carefully tuned and their best results are reported (except the offsets of FixedTime because of its random nature).

Conventional transportation baselines:

- **FixedTime**: Fixed-time with random offset [22]. Each phase has a fixed time of 15 seconds. For uni-directional traffic, there are only 2 phases (WE-straight, SN-straight). For traffic with turning vehicles, there are 4 phases.
- **GreenWave** [22]: This is the most classical method in transportation field to implement coordination that gives an optimal solution for unidirectional and uniform traffic on the arterial. It requires that all intersections share the same cycle length, which is the minimum value of the cycle length for individual intersections calculated using Webster’s theory [29]. The phase split percentage equals to the percentage between the demand of a designated phase and total demand. Offsets between intersections are equivalent to the free-flow travel time between two consecutive intersections.

Table 3: Data statistics of real-world traffic dataset

Dataset	Arrival rate (vehicles/h)				# of intersections
	Mean	Std	Max	Min	
Qingdao Rd., Jinan	3338.83	221.58	2748	3864	3
Beaver Ave., State College	2982.33	359.70	2724	3491	5
8-th Ave., NYC	6790.04	32.34	4968	7536	16
9-th Ave., NYC	4513.06	25.88	4416	6708	16
10-th Ave., NYC	6083.90	25.61	2892	5016	16
11-th Ave., NYC	4030.79	24.08	2472	4536	16

- **MaxPressure**: Max pressure control [27] is a state-of-the-art network-level traffic signal control method, which greedily chooses the phase with the maximum pressure, as introduced in Definition 5.3.

RL baselines:

- **LIT** is an individual deep reinforcement learning approach proposed in [35]. This method does not consider the traffic condition on downstream lanes in state and uses a reward with queue length.
- **GRL** is a coordinated reinforcement learning approach for multi-intersection control [25]. Specifically, the coordination is to design a coordination graph and to learn the joint local Q-function on two adjacent intersections directly.

6.3 Performance Comparison

Table 4 reports our experimental results using synthetic data under six-intersection arterial and real-world data *w.r.t.* average travel time. We have the following findings:

(1) Conventional transportation methods (FixedTime, GreenWave and MaxPressure) give poor performance. This is because the traffic in these settings is dynamic. Conventional methods, which

Table 4: Performance comparison between all the methods in the arterial with 6 intersections w.r.t. average travel time (the lower the better). Top-down: conventional transportation methods, learning methods, and our proposed method.

	Synthetic traffic				Real-world traffic					
	LightFlat	LightPeak	HeavyFlat	HeavyPeak	Qingdao Rd., Jinan	Beaver Ave., State College	8th Ave., NYC	9th Ave., NYC	10th Ave., NYC	11th Ave., NYC
FixedTime	93.29	109.50	325.48	246.25	317.40	336.29	432.60	469.54	347.05	368.84
GreenWave	98.39	124.09	263.36	286.85	370.30	332.06	451.98	502.30	317.02	314.08
MaxPressure	74.30	82.37	262.26	225.60	567.06	222.90	412.58	370.61	392.77	224.54
GRL	123.02	115.85	525.64	757.73	238.19	455.42	704.98	669.69	676.19	548.34
LIT	65.07	66.77	233.17	258.33	58.18	338.52	471.30	726.04	309.95	340.40
PressLight	**59.96**	**61.34**	**160.48**	**184.51**	**54.87**	**92.00**	**223.36**	**149.01**	**161.21**	**140.82**

Table 5: Detailed comparison of our proposed state and reward design and their effects w.r.t. average travel time (lower the better) under synthetic traffic data.

	HeavyFlat	HeavyPeak
LIT	233.17	258.33
LIT+out	201.56	281.21
LIT+out+seg	200.28	196.34
PressLight	**160.48**	**184.51**

rely heavily on over-simplified assumptions or prior knowledge on the traffic, may easily fail under the dynamic traffic scenarios.

(2) Our method PressLight outperforms all other RL methods. Though all the methods aim to learn to minimize the travel time, our reward design is proven to directly optimize towards it, while GRL and LIT are using mixed reward which may distract the model from learning efficiently.

(3) When the traffic grows larger (Config 3,4 to 1,2), PressLight becomes much better than other baselines. Under heavy traffic, a poor control strategy would make downstream queue may easily spill back and the green time would be wasted. The reward design of our agents considers balancing the queues on all the intersections within the arterial, which makes the performance even superior as the traffic becomes larger.

6.4 Study of PressLight

6.4.1 Effects of variants of our proposed method. We consider several variations of our model as follows.

- **LIT**. Instead of using the distribution of the vehicles, LIT simply uses phase and number of vehicles on each incoming lanes as its state (similar to LIT), and uses the reward defined same as LIT. This serves as a base model for later variants.
- **LIT+out**. Based on LIT, LIT+out adds the number of vehicles on outgoing lanes to its state, which has more information about its downstream intersections than LIT agents.
- **LIT+out+seg**. Based on LIT+out, LIT+out+seg uses the phase, the number of segments' vehicles on both incoming and outgoing lanes into its state, which is the same as our proposed state definition.
- **PressLight**. Our proposed method which further changes LIT+out+seg's reward to pressure.

Table 5 shows the performance of variants of our method:

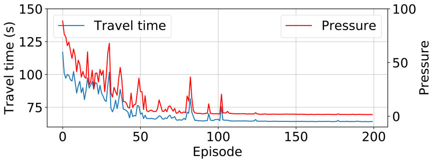


Figure 6: Convergence curve of average duration and our reward design (pressure). Pressure shows the same convergence trend with travel time.

(1) Giving the added state information (LIT+out and LIT+out+seg) boosts the performance. This makes sense since (1) LIT+out is able to observe traffic condition on outgoing lanes and helps to balance the queues for each intersection when there is congestion on outgoing lanes; (2) LIT+out+seg has the information about vehicle distributions which is the key factor for agents to learn the offsets.

(2) PressLight further outperforms LIT+out+seg owing to its reward definition. Instead of optimizing a reward that is not directly towards the travel time under arterial network, our reward design is proved to be a surrogate of average travel time. This demonstrates the effectiveness of our proposed reward design.

6.4.2 Average travel time related to pressure. Figure 6 illustrates the convergence curve of our agents learning process w.r.t. the average reward and the average pressure of each round. We can see that the travel time is closely correlated with pressure.

6.5 Performance on Mixed Scenarios

6.5.1 Heterogeneous intersections. We employ our model to two heterogeneous arterials, as is shown in Figure 7. For intersections with 3 legs, we use zero-padding to complete the state. For intersections with different lengths of lanes, our method can handle this well since the state is independent of the lane length. Table 6 illustrates the performance of our model against MaxPressure.

6.5.2 Arterials with a different number of intersections and network. We employ our model to arterials with 6, 10 and 20 intersections under synthetic data. As is shown in Table 6, our model could achieve better performance over conventional transportation method MaxPressure and reinforcement learning method LIT even when the number of intersections grows.

We also test our model a network with 9 intersections (3 × 3 grid). Table 6 shows the experiment results and we can see that PressLight can outperform MaxPressure and LIT under both traffic.

Table 6: Average travel time of different methods under arterials with a different number of intersections and network.

	6-intersection arterial		10-intersection arterial		20-intersection arterial		Grid network	
	HeavyFlat	HeavyPeak	HeavyFlat	HeavyPeak	HeavyFlat	HeavyPeak	HeavyFlat	HeavyPeak
MaxPressure	262.26	225.60	129.63	129.63	310.95	271.39	539.67	485.03
LIT	233.17	258.33	157.84	200.96	246.88	202.30	283.21	332.53
PressLight (ours)	**160.48**	**184.51**	**88.88**	**79.61**	**155.84**	**188.92**	**251.02**	**262.46**

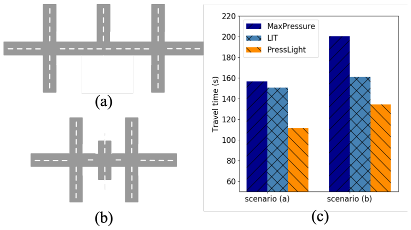
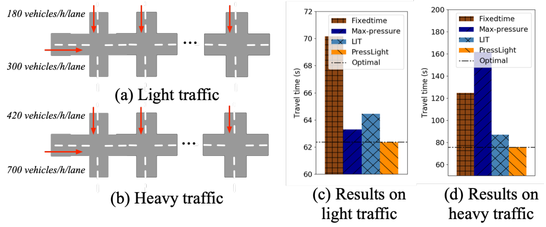


Figure 7: Average travel time of our method on heterogeneous intersections. (a) Different number of legs. (b) Different length of lanes. (c) Experiment results.



(a) Light traffic

(b) Heavy traffic

(c) Results on light traffic

(d) Results on heavy traffic

Figure 8: Performance comparison under uniform uni-directional traffic, where the optimal solution is known (GreenWave). Only PressLight can achieve the optimal.

6.6 Case Study

Another desirable property of PressLight is its ability to automatically coordinate the offset between adjacent intersections. To demonstrate this, we show two examples. Under simplified uniform traffic, we show that our model has learned the optimal solution which could be justified by transportation theories. Under the real-world traffic, the learned offset is visualized to reveal this property.

6.6.1 Synthetic traffic on the uniform, uni-directional flow. In this section, we perform experiments on the arterials with six homogeneous intersections under two traffic settings. One is for light traffic (arterial demand: 300 vehicle/hour/lane, side-street demand: 180 vehicle/hour/lane) and one is for heavy traffic (arterial demand: 700 vehicle/hour/lane, side-street demand: 420 vehicle/hour/lane). Both of them are uniform and uni-directional without turning traffic and two phases (*WE* for green light on arterial and *SN* for green light for side streets) are used for all intersections. Under these simplified scenarios, the optimal solution is known as GreenWave in transportation area as stated in [22]. As the optimal solution under these settings, GreenWave's policy includes the offsets between intersections and the phase split, which requires several prior knowledge to calculate them: The offset Δ equals to the block length l between two consecutive intersections divided by free-flow speed v; the optimal phase split ratio is equal to the ratio of the demand for a designated phase and total demand. In our experiments, $l \approx 300$ m, $v \approx 10$ m/s, hence, the optimal offset should be $\Delta \approx 30$ s, and the optimal phase split should be 1:0.6 (*WE*: *SN*).

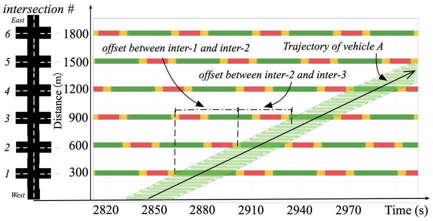


Figure 9: Offsets between intersections learnt by RL agents under uni-directional uniform traffic (700 vehicles/hour/lane on arterial)

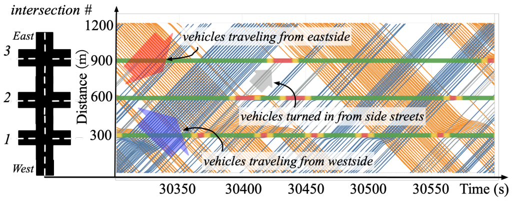


Figure 10: Space-time diagram with signal timing plan to illustrate the learned coordination strategy from real-world data on the arterial of Qingdao Road in the morning (around 8:30 a.m.) on August 6th.

Performance comparison. We compared PressLight with all aforementioned baselines and report their results in Figure 8. We can find that given GreenWave is the optimal solution, only our method PressLight achieves the same performance as GreenWave in both settings. This demonstrates that our RL agents can learn the optimal policy under these simplified scenarios.

Policy learned by RL agents. We use time-space diagrams to show the trajectories of vehicles and phase plans of traffic signal controllers. In a time-space diagram like Figure 9, the x-axis is the time and the y-axis is the distance (from a reference point, here we use the westernmost point as the reference point). As it is shown in Figure 9, there are six bands with green-yellow-red colors indicating the changing phases of six intersections. The black line with an arrow is the trajectory of a vehicle, where the x-axis tells the time and the y-axis tells the location. Vehicles that travel within the green dashed area will experience a green wave. For example, vehicle *A* enters the system at 2850 second and traveled through 5 intersections at 3000 second, experiencing consecutive green lights during its trip. The slope indicates the speed of the vehicle.

We have several observations: (1) PressLight can learn the optimal phase split as GreenWave. As is shown in Figure 9, our method learns optimal phase split (approximately 1:0.6, with 25 seconds of

WE, 15 seconds of *SN*, and 10 seconds of yellow light). (2) PressLight can learn the optimal offset and form a green wave. In Figure 9, the offset is approximately 30s between two consecutive traffic signals and a green wave can be seen (dashed green area in Figure 9). This demonstrates that our RL method can learn the optimal policy given by GreenWave.

6.6.2 Real-world traffic in Jinan. In this section, we make observations on the policies we learned from the real data for the arterial of Qingdao Road (*East* and *West* direction) during the morning peak hour (around 8:30 a.m.) on August 6th. In Figure 10, a time-space diagram is drawn with time on the horizontal axis and distance (from a reference point, here we use the westernmost point on the arterial as the reference point) on the vertical axis. Most of the blue and orange lines are straight, indicating most vehicles on the arterial are not stopped by red lights, which means our method can automatically form a green wave.

7 CONCLUSION

In this paper, we propose a novel RL method for multi-intersection traffic signal control on the arterials. We conduct extensive experiments using both synthetic and real data and demonstrate the superior performance of our method over the state-of-the-art. Specifically, we draw a connection on the design between reinforcement learning with conventional transportation control methods. It is also the first time the individual RL model automatically achieves coordination along arterial without any prior knowledge.

We acknowledge the limitations of our model and would like to point out several future directions. In our experiment, we did not model the behavior of vehicles. The behavior of vehicles (e.g., routing) in the real-world may change dynamically in response to traffic lights. Another direction can be reducing the cost of learning. Since RL is learning from trial-and-error, deploying an online updated RL model in real-world could be dangerous and costly.

ACKNOWLEDGMENTS

The work was supported in part by NSF awards #1652525, #1618448, and #1639150. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

REFERENCES

[1] Monireh Abdoos, Nasser Mozayani, and Ana LC Bazzan. 2013. Holonic multi-agent system for traffic signals control. *Engineering Applications of Artificial Intelligence* 26, 5 (2013), 1575–1587.

[2] Baher Abdulhai, Rob Pringle, and Grigoris J Karakoulas. 2003. Reinforcement learning for true adaptive traffic signal control. *Journal of Transportation Engineering* 129, 3 (2003), 278–285.

[3] Baher Abdulhai, Rob Pringle, and Grigoris J Karakoulas. 2003. Reinforcement learning for true adaptive traffic signal control. *Journal of Transportation Engineering* 129, 3 (2003), 278–285.

[4] Itamar Arel, Cong Liu, T Urbanik, and AG Kohls. 2010. Reinforcement learning-based multi-agent system for network traffic signal control. *IET Intelligent Transport Systems* 4, 2 (2010), 128–135.

[5] Florence Boillot, Sophie Midenet, and Jean-Claude Pierrelee. 2006. The real-time urban traffic control system CRONOS: Algorithm and experiments. *Transportation Research Part C: Emerging Technologies* 14, 1 (2006), 18–38.

[6] Tim Brys, Tong T Pham, and Matthew E Taylor. 2014. Distributed learning and multi-objectivity in traffic light control. *Connection Science* 26, 1 (2014), 65–83.

[7] Noe Casas. 2017. Deep Deterministic Policy Gradient for Urban Traffic Light Control. *arXiv preprint arXiv:1703.09035* (2017).

[8] ALCB Bruno Castro da Silva, Denise de Oliveria, and EW Basso. 2006. Adaptive traffic control with reinforcement learning. In *Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 80–86.

[9] Samah El-Tantawy and Baher Abdulhai. 2010. An agent-based learning towards decentralized and coordinated traffic signal control. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC* (2010), 665–670. https://doi.org/10.1109/ITSC.2010.5625066

[10] Samah El-Tantawy, Baher Abdulhai, and Hossam Abdelgawad. 2013. Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (MARLIN-ATSC): methodology and large-scale application on downtown Toronto. *IEEE Transactions on Intelligent Transportation Systems* 14, 3 (2013), 1140–1150.

[11] Nathan H Gartner. 1983. *OPAC: A demand-responsive strategy for traffic signal control.* Number 906.

[12] Jean-Jacques Henry, Jean Loup Farges, and J Tuffal. 1984. The PRODYN real time traffic algorithm. In *Control in Transportation Systems*. Elsevier, 305–310.

[13] Lior Kuyer, Shimon Whiteson, Bram Bakker, and Nikos Vlassis. 2008. Multiagent reinforcement learning for urban traffic control using coordination graphs. *Machine learning and knowledge discovery in databases* (2008), 656–671.

[14] Li Li, Yisheng Lv, and Fei-Yue Wang. 2016. Traffic signal timing via deep reinforcement learning. *IEEE/CAA Journal of Automatica Sinica* 3, 3 (2016), 247–254.

[15] Xiaoyuan Liang, Xunsheng Du, Guiling Wang, and Zhu Han. 2018. Deep reinforcement learning for traffic light control in vehicular networks. *arXiv preprint arXiv:1803.11115* (2018).

[16] Jennie Lioris, Alex Kurzhanskiy, and Pravin Varaiya. 2013. Adaptive Max Pressure Control of Network of Signalized Intersections. *Transportation Research Part C* 36, 22 (2013), 177–195.

[17] John DC Little, Mark D Kelson, and Nathan H Gartner. 1981. MAXBAND: A versatile program for setting signals on arteries and triangular networks. (1981).

[18] Patrick Mannion, Jim Duggan, and Enda Howley. 2016. An experimental review of reinforcement learning algorithms for adaptive traffic signal control. In *Autonomic Road Transport Support Systems*. Springer, 47–66.

[19] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (2015), 529.

[20] Seyed Sajad Mousavi, Michael Schukat, Peter Corcoran, and Enda Howley. 2017. Traffic Light Control Using Deep Policy-Gradient and Value-Function Based Reinforcement Learning. *arXiv preprint arXiv:1704.08883* (2017).

[21] Dennis I Robertson. 1969. TRANSYT: a traffic network study tool. (1969).

[22] Roger P Roess, Elena S Prassas, and William R Mcshane. 2011. *Traffic Engineering.* Pearson/Prentice Hall.

[23] Suvrajeet Sen and K Larry Head. 1997. Controlled optimization of phases at an intersection. *Transportation science* 31, 1 (1997), 5–17.

[24] Thomas Urbanik, Alison Tanaka, Bailey Lozner, Eric Lindstrom, Kevin Lee, Shaun Quayle, Scott Beaird, Shing Tsoi, Paul Ryus, Doug Gettman, et al. 2015. *Signal timing manual.* Transportation Research Board.

[25] Elise Van der Pol and Frans A Oliehoek. 2016. Coordinated deep reinforcement learners for traffic light control. *Proceedings of Learning, Inference and Control of Multi-Agent Systems (at NIPS 2016).*

[26] Pravin Varaiya. 2013. Max pressure control of a network of signalized intersections. *Transportation Research Part C: Emerging Technologies* 36 (2013), 177–195.

[27] Pravin Varaiya. 2013. Max pressure control of a network of signalized intersections. *Transportation Research Part C: Emerging Technologies* 36 (2013), 177–195.

[28] Pravin Varaiya. 2013. *The Max-Pressure Controller for Arbitrary Networks of Signalized Intersections.* Vol. 2. https://doi.org/10.1007/978-1-4614-6243-9

[29] F. V Webster. 1958. Traffic signal settings. *Road Research Technical Paper* 39 (1958).

[30] Hua Wei, Guanjie Zheng, Vikash Gayah, and Zhenhui Li. 2019. A Survey on Traffic Signal Control Methods. *CoRR* abs/1904.08117 (2019). arXiv:1904.08117

[31] Hua Wei, Guanjie Zheng, Huaxiu Yao, and Zhenhui Li. 2018. IntelliLight: A Reinforcement Learning Approach for Intelligent Traffic Light Control. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2496–2505.

[32] MA Wiering. 2000. Multi-agent reinforcement learning for traffic light control. In *Machine Learning: Proceedings of the Seventeenth International Conference (ICML'2000)*. 1151–1158.

[33] Huichu Zhang, Siyuan Feng, Chang Liu, Yaoyao Ding, Yichen Zhu, Zihan Zhou, Weinan Zhang, Yong Yu, Haiming Jin, and Zhenhui Li. 2019. CityFlow: A Multi-Agent Reinforcement Learning Environment for Large Scale City Traffic Scenario. (2019).

[34] Guanjie Zheng, Yuanhao Xiong, Xinshi Zang, Jie Feng, Hua Wei, Huichu Zhang, Yong Li, Kai Xu, and Zhenhui Li. 2019. Learning Phase Competition for Traffic Signal Control. *CoRR* abs/1905.04722 (2019). arXiv:1905.04722

[35] Guanjie Zheng, Xinshi Zang, Nan Xu, Hua Wei, Zhengyao Yu, Vikash Gayah, Kai Xu, and Zhenhui Li. 2019. Diagnosing Reinforcement Learning for Traffic Signal Control. *CoRR* abs/1905.04716 (2019). arXiv:1905.04716