A Survey on Traffic Signal Control Methods

HUA WEI, College of Information Sciences and Technology, Penn State University, USA
GUANJIE ZHENG, College of Information Sciences and Technology, Penn State University, USA
VIKASH GAYAH, Department of Civil Engineering, Penn State University, USA
ZHENHUI LI, College of Information Sciences and Technology, Penn State University, USA

Traffic signal control is an important and challenging real-world problem, which aims to minimize the travel time of vehicles by coordinating their movements at the road intersections. Current traffic signal control systems in use still rely heavily on oversimplified information and rule-based methods, although we now have richer data, more computing power and advanced methods to drive the development of intelligent transportation. With the growing interest in intelligent transportation using machine learning methods like reinforcement learning, this survey covers the widely acknowledged transportation approaches and a comprehensive list of recent literature on reinforcement for traffic signal control. We hope this survey can foster interdisciplinary research on this important topic.

Additional Key Words and Phrases: traffic signal control, transportation, reinforcement learning, deep learning, mobility data

ACM Reference Format:
Hua Wei, Guanjie Zheng, Vikash Gayah, and Zhenhui Li. 2020. A Survey on Traffic Signal Control Methods. 1, 1 (January 2020), 32 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Traffic congestion is a growing problem that continues to plague urban areas with negative outcomes to both the traveling public and society as a whole. These negative outcomes will only grow over time as more people flock to urban areas. In 2014, traffic congestion costs Americans over $160 billion in lost productivity and wasted over 3.1 billion gallons of fuel [Economist 2014]. Traffic congestion was also attributed to over 56 billion pounds of harmful CO2 emissions in 2011 [Schrank et al. 2015]. In the European Union, the cost of traffic congestion was equivalent to 1% of the entire GDP [Schrank et al. 2012]. Mitigating congestion would have significant economic, environmental and societal benefits. Signalized intersections are one of the most prevalent bottleneck types in urban environments, and thus traffic signal control plays a vital role in urban traffic management.

1.1 Current Situation

In many modern cities today, the widely-used adaptive traffic signal control systems such as SCATS [Lowrie 1992] and SCOOT [Hunt et al. 1982, 1981] heavily rely on manually designed traffic signal plans. Such manually set traffic signal plans are designed to be dynamically selected according to the traffic volume detected by loop

Authors' addresses: Hua Wei, College of Information Sciences and Technology, Penn State University, University Park, PA, 16802, USA, hzw77@ist.psu.edu; Guanjie Zheng, College of Information Sciences and Technology, Penn State University, University Park, PA, 16802, USA, gjz5038@ist.psu.edu; Vikash Gayah, Department of Civil Engineering, Penn State University, University Park, PA, 16802, USA, gayah@engr.psu.edu; Zhenhui Li, College of Information Sciences and Technology, Penn State University, University Park, PA, 16802, USA, jessieli@ist.psu.edu.

XXXX-XXXX/2020/1-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

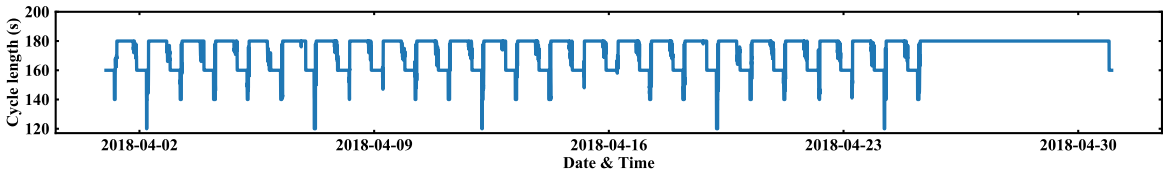

Fig. 1. Traffic signal timing of a downtown intersection in Hangzhou, China. The x- and y-axis indicate the time and cycle length of the traffic signal. The cycle length rarely changes as time goes by.

se However, many intersections do not have loop sensors installed or the loop sensors are poorly maintained. Moreover, the loop sensors are activated only when vehicles pass through them; thus, they can only provide partial information about the vehicle through them. As a result, the signal cannot perceive and react to real-time traffic patterns. Engineers need to manually change the traffic signal timings in the signal control system under certain traffic condition scenarios. Figure 1 shows the traffic signal timing at an intersection in a city of China, and the traffic signal timing rarely changes regardless of the real traffic changes throughout the day.

1.2 Opportunities

First, today we have much richer information that can be collected from various sources. Traditional traffic signal control relies on data from loop sensors, which can only sense the vehicle passing. However, new data sources are quickly becoming available that can serve as input for traffic signal control purposes. For instance, street-facing surveillance cameras used for security purposes can also provide a more detailed depiction of the traffic situation on nearby roads, specifically on how many cars are waiting in the lane, how many cars are taking turns, where they are located, and how fast they are traveling. In addition, large-scale trajectory data can be collected from various sources such as navigation applications (e.g., Google Maps), ride-sharing platforms (e.g., Uber), and GPS-equipped vehicles that share information with the nearby infrastructure (e.g., connected vehicles). Such data provide us with more insight into how vehicles arrive at intersections. We have reached a stage of sufficient mobility information that can describe the traffic dynamics in the city more clearly, which is an essential resource for us to improve the traffic control system.

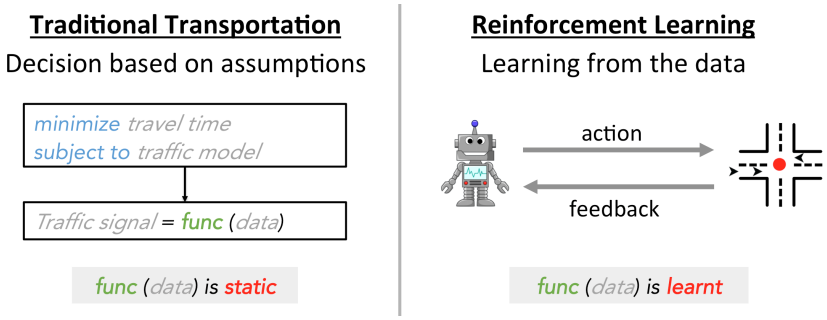

Fig. 2. Difference between traditional transportation approach and machine learning approach.

Second, today we have the much stronger computing power and advanced computational models. The typical approach that transportation researchers take is to cast traffic signal control as an optimization problem under certain assumptions about the traffic model, e.g., vehicles come in a uniform and constant rate [Roess et al.

2004]. Various (and sometimes strong) assumptions have to be made in order to make the optimization problem tractable. The key issue here is that these assumptions deviate from the real world and often do so significantly. As we know, real-world traffic condition evolves in a complicated way, affected by many factors such as driver's preference, interactions with vulnerable road users (e.g., pedestrians, cyclists, etc.), weather and road conditions. These factors can hardly be fully described in a traffic model.

On the other hand, machine learning techniques can directly learn from the observed data without making unrealistic assumptions about the model. However, typical supervised learning does not apply here because existing traffic signal control systems follow pre-defined signal plans so we do not have enough training data to differentiate good and bad traffic signal plan strategies. Instead, we have first to take actions to change the signal plans and then learn from the outcomes. This trial-and-error approach is also the core idea of reinforcement learning (RL). In essence, an RL system generates and executes different strategies (e.g., for traffic signal control) based on the current environment. It will then learn and adjust the strategy based on the feedback from the environment. This reveals the most significant difference between transportation approaches and our RL approaches, which is illustrated in Figure 2: in traditional transportation research, the model $func(data)$ is static; in reinforcement learning, the model is dynamically learned through trial-and-error in the real environment.

1.3 Motivation of This Survey

With the surge of AI technology and increasingly available city data, governments and industries are now actively seeking solutions to improve the transportation system. For example, in China, Alibaba and Didi Chuxing are working on using mobility data and advanced computing technology to enhance city transportation [CNN 2019; Wire 2018]. This survey could provide a useful reference for the industry when they revolutionize current traffic signal control systems by trying out the RL-based methods. Specifically, we discuss the learning approaches of recent RL-based methods with their weaknesses and strengths and benchmark the experiment settings of the existing works.

At the same time, with the recent success in reinforcement learning techniques, we see an increasing interest in academia to use reinforcement learning to improve traffic signal control [Mannion et al. 2016]. However, most existing machine learning approaches tend to ignore classic transportation approaches and lack a good comparison with existing transportation approaches. This survey takes a comprehensive view of both machine learning and transportation engineering and hopes to facilitate this interdisciplinary research direction.

1.4 Scope of This Survey

In this survey, we will cover many classical or widely accepted transportation approaches in traffic signal control. We refer readers interested in comprehensive transportation approaches to [Li et al. 2014; Papageorgiou et al. 2003; Roess et al. 2004]. The application of reinforcement learning methods in traffic signal control is relatively new. Recent advances in deep reinforcement learning also arouses new applications of RL in traffic signal control problem. While [Yau et al. 2017] and [Mannion et al. 2016] provide comprehensive surveys mainly on earlier studies before the popularity of deep reinforcement learning, in this survey, we will have comprehensive coverage in RL-based traffic signal control approaches, including the recent advances in deep RL-based traffic signal control methods.

2 PRELIMINARY

2.1 Term Definition

Terms on road structure and traffic movement:

- **Approach**: A roadway meeting at an intersection is referred to as an approach. At any general intersection, there are two kinds of approaches: incoming approaches and outgoing approaches. An incoming approach

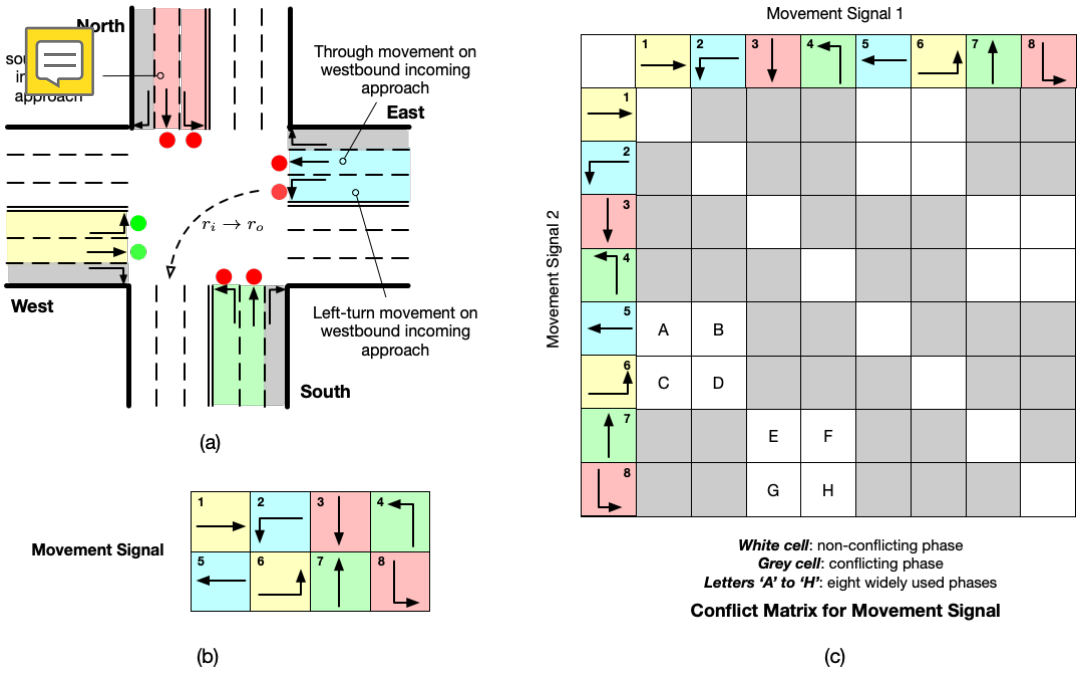

Fig. 3. Definitions of traffic movement and traffic signal phases.

is one on which cars can enter the intersection; an outgoing approach is one on which cars can leave the intersection. Figure 3(a) shows a typical intersection with four incoming approaches and outgoing approaches. The southbound incoming approach is denoted in this figure as the approach on the north side in which vehicles are traveling in the southbound direction.

- **Lane**: An approach consists of a set of lanes. Similar to approach definition, there are two kinds of lanes: incoming lanes and outgoing lanes (also known as approaching/entering lane and receiving/exiting lane in some references [El-Tantawy and Abdulhai 2012; Stevanovic 2010]).
- **Traffic movement**: A traffic movement refers to vehicles moving from an incoming approach to an outgoing approach, denoted as $(r_i \rightarrow r_o)$, where r_i and r_o is the incoming lane and the outgoing lane respectively. A traffic movement is generally categorized as left turn, through, and right turn.

Terms on traffic signal:

- **Movement signal**: 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. For the four-leg intersection shown in Figure 3(a), the right-turn traffic can pass regardless of the signal, and there are eight movement signals in use, as shown in Figure 3(b).
- **Phase**: A phase is a combination of movement signals. Figure 3(c) shows the conflict matrix of the combination of two movement signals in the example in Figure 3(a) and Figure 3(b). The grey cell indicates the corresponding two movements conflict with each other, i.e. they cannot be set to 'green' at the same time (e.g., signals #1 and #2). The white cell indicates the non-conflicting movement signals. All the non-conflicting signals will generate eight valid paired-signal phases (letters 'A' to 'H' in Figure 3(c)) and eight single-signal phases (the diagonal cells in conflict matrix). Here we letter the paired-signal phases only

because in an isolated intersection, it is always more efficient to use paired-signal phases. When considering multiple intersections, single-signal phase might be necessary because of the potential spill back.

- **Phase sequence**: A phase sequence is a sequence of phases which defines a set of phases and their order of changes.
- **Signal plan**: A signal plan for a single intersection is a sequence of phases and their corresponding starting time. Here we denote a signal plan as $(p_1, t_1)(p_2, t_2)\dots(p_i, t_i)\dots$, where p_i and t_i stand for a phase and its starting time.
- **Cycle-based signal plan**: A cycle-based signal plan is a kind of signal plan where the sequence of phases operates in a cyclic order, which can be denoted as $(p_1, t_1^1)(p_2, t_2^1)\dots(p_N, t_N^1)(p_1, t_1^2)(p_2, t_2^2)\dots(p_N, t_N^2)\dots$, where $p_1, p_2, \dots, p_N$ is the repeated phase sequence and t_i^j is the starting time of phase p_i in the j-th cycle. Specifically, $C^j = t_1^{j+1} - t_1^j$ is the cycle length of the j-th phase cycle, and $\{\frac{t_2^j - t_1^j}{C^j}, \dots, \frac{t_N^j - t_{N-1}^j}{C^j}\}$ is the phase split of the j-th phase cycle. Existing traffic signal control methods usually repeats similar phase sequence throughout the day.

2.2 Objective

The objective of traffic signal control is to facilitate safe and efficient movement of vehicles at the intersection. Safety is achieved by separating conflicting movements in time and is not considered in more detail here. Various measures have been proposed to quantify efficiency of the intersection from different perspectives:

- Travel time. In traffic signal control, travel time of a vehicle is defined as the time different between the time one car enters the system and the time it leaves the system. One of the most common goals is to minimize the average travel time of vehicles in the network.
- Queue length. The queue length of the road network is the number of queuing vehicles in the road network.
- Number of stops. The number of stops of a vehicle is the total times that a vehicle experienced.
- Throughput. The throughput is the number of vehicles that have completed their trip in the road network during a period.

2.3 Special Considerations

In practice, additional attention should be paid to the following aspects:

(1) Yellow and all-red time. A yellow signal is usually set as a transition from a green signal to a red one. Following the yellow, there is an all-red period during which all the signals in an intersection are set to red. The yellow and all-red time, which can last from 3 to 6 seconds, allow vehicles to stop safely or pass the intersection before vehicles in conflicting traffic movements are given a green signal.
(2) Minimum green time. Usually, a minimum green signal time is required to ensure pedestrians moving during a particular phase can safely pass through the intersection.
(3) Left turn phase. Usually, a left-turn phase is added when the left-turn volume is above certain threshold.

3 METHODS IN TRANSPORTATION ENGINEERING

In this section, we introduce several selected classical methods in transportation field, which should be taken into comparison as baselines for RL-based methods. An overview of covered methods is shown in Table 1. For a more comprehensive survey for the methods in transportation, we refer the interested readers to [Martinez et al. 2011; Roess et al. 2004].

Table 1. Overview of classic optimization-based transportation methods

Method	Prior Knowledge	Data Input	Output
Webster	phase sequence in a cycle	traffic volume	cycle-based signal plan for an individual intersection
GreenWave, Maxband	cycle-based signal plan at individual intersections	traffic volume, speed limit, lane length	offset for a cycle-based signal plan
Actuated, SOTL	phase sequence, phase change rule	traffic volume	change to next phase according to the rule and data
Max-pressure	none	queue length	signal plan for all intersections
SCATS	signal plan for all intersections	traffic volume	adjusted signal plans

3.1 Webster (single intersection)

For a single (isolated) intersection, the traffic signal plan in transportation engineering usually consists of a pre-timed cycle length , fixed cycle-based phase sequence, and phase split. Webster method [Koonce et al. 2008] is one of the widely-used method in field to calculate the cycle length and phase split for a single intersection. Assuming the traffic flow is uniform during a certain period (i.e., past 5 or 10 minutes), it has a closed-form solution shown in Eq. (1) and (2) to generate the optimal cycle length and phase split for a single intersection that minimizes the travel time of all vehicles passing the intersection.

The calculation of the desired cycle length C_{des} relies on Webster's Equation:

$$C_{des}(V_c) = \frac{N \times t_L}{1 - \frac{V_c}{\frac{3600}{h} \times PHF \times (v/c)}} \tag{1}$$

where N is the number of phases; t_L is the total loss time per phase, which can be treated as a parameter related to the all-red time and the acceleration and deceleration of vehicles; parameter h is saturation headway time (seconds/vehicle), which is the smallest time interval between successive vehicles passing a point; PHF is short for peak hour factor, which is a parameter measuring traffic demand fluctuations within the peak hour; and the parameter v/c is desired volume-to-capacity ratio, which indicates how busy the intersection is in a signal timing context. These parameters usually vary in different traffic conditions and are usually selected based on empirical observations and agency standards. The equation represents a function of V_c, where $V_c = \Sigma_i^N V_c^i$ is the sum of all critical lane volumes, which indicates how busy the intersection is in the signal timing context, and V_c^i stands for the critical lane volumes for phase i, where a critical lane is the approaching lane with the highest ratio of traffic flow to saturation flow in a phase, usually indicated by the queue length.

Once the cycle length is decided, the green split (i.e., the green time over the cycle length) is then calculated to be proportional to the ratios of critical lane volumes served by each phase, as indicated in Eq. (2):

$$\frac{t_i}{t_j} = \frac{V_c^i}{V_c^j} \tag{2}$$

where t_i and t_j stands for the phase duration for phase i and j.

Eq. (1) and (2) are typically applied using aggregated data to develop fixed-time plans. However, these equations can also be modified for real-time application. For example, when the traffic is uniform, the Webster method can be proved to minimize the travel time of all vehicles passing the intersection or maximize the intersection capacity. By collecting data for a short time period and assuming no fluctuation in traffic demand and no redundancy left in a cycle, i.e., setting both PHF and v/c to 1, Eq. (2) can be re-organized as

$$(1 - \frac{N \times t_L}{C_{des}})\frac{3600}{h} = V_c. \tag{3}$$

The left side of Eq. (3) is the proportion of time utilized for traffic movements at the intersection multiplied by the saturation flow of vehicles, which is the capacity of the intersection. When capacity equals to the traffic demand V_c, cycle length C_{des} is the minimum value that tightly satisfies the traffic demands. For a given traffic demand, if a signal cycle length smaller than C_{des} is applied, the queue length will keep increasing and the intersection gets over-saturated. If a signal cycle length larger than C_{des} is applied, the mean delay for each vehicle at the intersection will grow linearly with the cycle length.

3.2 GreenWave

While Webster method generates the signal plan for a single intersection, the offsets (i.e., starting time between phase cycles at adjacent intersections) between the signal timings for adjacent traffic signals should also be optimized as signals are often in close proximity. Failure to do so can lead to decisions being made at one signal that can deteriorate traffic operations at another.

GreenWave [Roess et al. 2004] is the most classical method in transportation field to implement coordination, which aims to optimize the offsets to reduce the number of stops for vehicles traveling along one certain direction. Given the signal plan (i.e., cycle length and phase split) of individual intersections, GreenWave requires all intersections to share the same cycle length, which is the maximum value of the given cycle lengths for individual intersections. The offsets between intersections are calculated by the following equation:

$$\Delta t_{i,j} = \frac{L_{i,j}}{v} \tag{4}$$

where $L_{i,j}$ is the road length between intersection i and j, and v is the expected travel speed of vehicles on the road.

This method can form a green wave along the designated direction of traffic where vehicles traveling along that direction can benefit from a progressive cascade of green signals without stopping at any intersections. However, GreenWave only optimizes for unidirectional traffic.

3.3 Maxband

Another typical approach, Maxband [Little et al. 1981], also takes the signal plans for individual intersections as input and optimizes the offsets for adjacent traffic signals. Different from GreenWave, it aims to reduce the number of stops for vehicles traveling along *two* opposite directions through finding a maximal bandwidth based on the signal plans of individual intersections within the system. The bandwidth for one direction is a portion of time that the synchronized green wave lasts in a cycle length. A larger bandwidth implies that more traffic along one direction can progress through the signals without stops.

Like GreenWave, Maxband [Little et al. 1981] requires all intersections to share the same cycle length, which equals to the maximum value of all cycle lengths of intersections. Then Maxband formulates a mixed integer linear programming model to generate a symmetric, uniform-width bandwidth, subject to the following physical constraints:

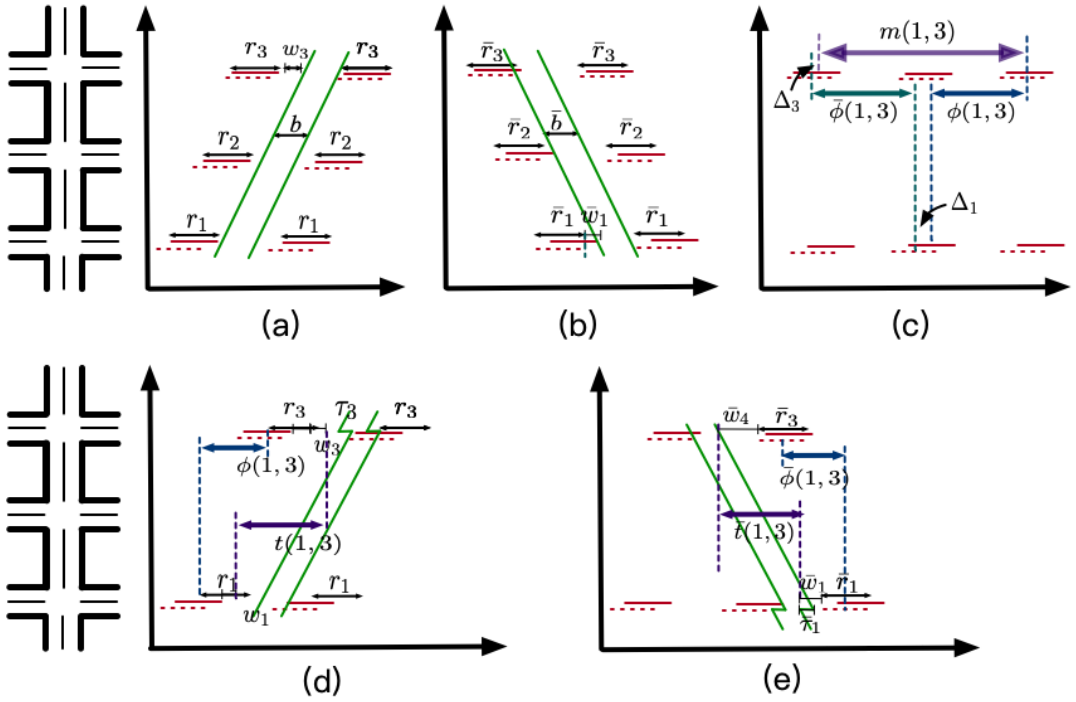

Fig. 4. Illustration of the constraints in Maxband under a three-intersection arterial network. Red solid/dotted lines indicate the red signal for inbound/outbound direction. The green band indicates the green wave. (a) - (b): Bandwidth constraints of each individual intersection on inbound and outbound direction. (c): Temporal constraint between two intersections. (d) - (e): Spatial constraints between two intersections.

- Constraints on the bandwidth of individual intersections. For simplicity, all the time intervals in the following refers to the ratio of time over the cycle length C.
  - For each direction (here we use inbound and outbound to differentiate two designated directions), the green time should be greater than the bandwidth. We have:

$$w_i + b \leq 1 - r_i, \ w_i > 0 \tag{5a}$$

$$\bar{w}_i + \bar{b} \leq 1 - \bar{r}_i, \ \bar{w}_i > 0 \tag{5b}$$

    Here, $w_i/\bar{w}_i$ denotes the time interval between the end of red time and the start of bandwidth on inbound/outbound direction, b is the bandwidth variable, and $r_i/\bar{r}_i$ is the red time on inbound/outbound direction.
  - For two different directions (inbound and outbound), usually the bandwidths for inbound and outbound are set equal:

$$b = \bar{b} \tag{6}$$

- Temporal and spatial constraints on offsets between two intersections i and j.
  - Temporal constraints.

$$\phi(i,j) + \bar{\phi}(i,j) + \Delta_i - \Delta_j = m(i,j) \tag{7}$$

Here, $\phi/\bar{\phi}$ is the inbound/outbound offset between intersections, Δ is the intra-intersection offset between the inbound and outbound red time, $m(i,j)$ is an integer variable, indicating a multiple of cycle length.

– Spatial constraints. The travel time of a vehicle starting from one intersection to another satisfies a function of the offset between them, the queue clearance time at the destined intersection and their red time.

$$\phi(i,j) + 0.5 * r_j + w_j + \tau_j = 0.5 * r_i + w_i + t(i,j) \tag{8a}$$

$$\bar{\phi}(i,j) + 0.5 * \bar{r}_j + \bar{w}_j = 0.5 * \bar{r}_i + \bar{w}_i - \bar{\tau}_i + \bar{t}(i,j) \tag{8b}$$

Here, $t/\bar{t}$ is the travel time inbound/outbound between intersections, which is relevant to the road length and traffic speed. $\tau/\bar{\tau}$ is the queue clearance time at an intersection, which is relevant to the turn-in and generated traffic.

In summary, Maxband aims to find b, $\bar{b}$ w_i, $\bar{w}_i$, m_i that:

$$\begin{aligned} &maximize \;\; b \\ &subject\ to \;\; Eq.\ (5a),\ (5b),\ (6),\ (7),\ (8a),\ (8b) \\ &\qquad\qquad m_i \in \mathbb{N} \\ &\qquad\qquad b,\ \bar{b},\ w_i,\ \bar{w}_i \geq 0, i = 1, \ldots, n \end{aligned} \tag{9}$$

A number of significant extensions of Maxband have been introduced based on the original method [Little et al. 1981] in order to consider a variety of new aspects. [Gartner et al. 1991] extends to include asymmetric bandwidths in the opposing direction, variable left-turn phase sequence, as well as decisions on cycle time length and link specific progression speeds. [Stamatiadis and Gartner 1996] presents the new multi-band, multi-weight approach, which also incorporates all previous decision capabilities.

3.4 Actuated Control

Actuated control measures the "requests" for a green signal from the current phase and other competing phases, then based on some rules to decide whether to keep or change the current phase. The definitions of "request" for the current phase and other competing phases are:

Table 2. Rules for actuated control

Request from current phase?	Request from other phases?	Action
Yes	Yes	Fully-actuated control:If the duration for the current phase is larger than a threshold, change to the next phase; otherwise, keep the current phase Semi-actuated control: Keep the current phase
	No	Keep the current phase
No	Yes	Change to the next phase
	No	Fully-actuated: Keep the current phase Semi-actuated: Change to the default phase (usually set as green signal for the main road

- The request on the current phase to extend the green signal time is generated when the duration for the current phase does not reach a minimum time period, or there is a vehicle on the incoming lane of the current phase and is within close distance to the intersection. And we call this vehicle is "approaching the green signal" in short.
- The request on the competing phases for a green signal is generated when the number of waiting vehicles in the competing phases is larger than a threshold.

Based on the difference in rules, there are two kinds of actuated control: fully-actuated control and semi-actuated control. The rules for the decision of keep or changing the current phase are listed in Table 2.

3.5 Self-organizing Traffic Light Control

Table 3. Rules for SOTL control

Request from current phase?	Request from competing phases?	Action
Yes	Yes	Keep the current phase
	No	
No	Yes	Change to the next phase
	No	Keep the current phase

Self-Organizing Traffic Light Control (SOTL) is basically the same as fully-actuated control with additional demand responsive rules [Cools et al. 2013; Gershenson 2004]. The main difference between SOTL and fully-actuated control is on the definition of request on the current phase (although they both require a minimum green phase duration): in fully-actuated control, the request on the current phase will be generated whenever there is a vehicle approaching the green signal, while in SOTL, the request will not be generated unless the number of vehicles approaching the green signal is larger than a threshold which is not necessarily one. Specifically, the rules for SOTL control are shown in Table 3.

3.6 Max-pressure

Max-pressure control [Varaiya 2013] aims to reduce the risk of over-saturation by balancing queue length between neighboring intersections by minimizing the "pressure" of the phases for an intersection. The concept of pressure is illustrated in Figure 5. Formally, the pressure of a *movement signal* can be defined as the number of vehicles on incoming lanes (of the traffic movement) minus the number of vehicles on the corresponding outgoing lanes; the pressure of a *phase* is defined as the difference between the total queue length on incoming approaches and outgoing approaches. By setting the objective as minimizing the pressure of phases for individual intersections, Max-pressure is proved to maximize the throughput of the whole road network. For readers interested in the proof, please refer to [Varaiya 2013].

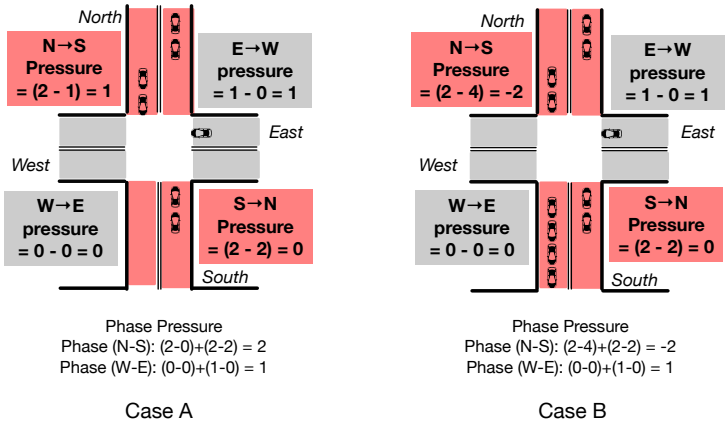

Fig. 5. Illustration of max pressure control in two cases. In both cases, there are four movement signals: North→South, South→North, East→West and West→East and there are two phases: $Phase(N-S)$ which sets green signal in the North→South and South→North direction, and $Phase(N-S)$ which sets green signal in the East→West and West→East direction. In Case A, Max-pressure selects $Phase(N-S)$ since the pressure of $Phase(N-S)$ is higher than $Phase(W-E)$; in Case B, Max-pressure selects $Phase(W-E)$.

Max-pressure control proposed in [Varaiya 2013] is formally summarized in Algorithm 1. From line 3 to line 7, this method selects the phase with the maximum pressure, activates it as the next phase and keeps the selected phase for a given period of time t_{min}.

Algorithm 1: Max-pressure Control

Input: Current phase time t, minimum phase duration length t_{min}

1 **forall** *timestamp* **do**
2 $t = t + 1$
3 **if** $t >= t_{min}$ **then**
4 Calculate the pressure P_i for each phase i;
5 Set the next phase as $\arg\max_i\{P_i\}$;
6 $t = 0$;
7 **end**
8 **end**

3.7 SCATS

SCATS [Lowrie 1990] (Sydney Coordinated Adaptive Traffic System) takes pre-defined signal plans (i.e., cycle length, phase split and offsets) as input and iteratively selects from these traffic signals according to the pre-defined performance measurements. Its measurement like the degree of saturation (DS) is detailed as follows:

$$DS = \frac{g_E}{g} = \frac{g - (h' - h \times n)}{g} \tag{10}$$

where g is the available green signal time (in seconds), g_E is the effective green time during which there are vehicles passing through the intersection and equals to available green signal time minus the wasted green time; and the wasted green time is calculated through the detection - h' is the detected total time gap, n is the

detected number of vehicles, and h is the unit saturation headway (seconds) between vehicles, which stands for the smallest time interval between successive vehicles passing a point and usually set with expert knowledge.

The phase split (i.e., the phase time ratio of all phases), cycle length and offsets are selected from several pre-defined plans using similar mechanisms. Take the selection mechanism of the phase split as an example. As shown in Algorithm 2, the algorithm first calculates DS for the current signal plan using Eq. (10) at the end of each cycle (line 2 to 4). Then from line 6 to line 10, the algorithm infers the DS of other signal plans which are not applied in the current cycle using the following equation:

$$\bar{DS}_p^j = \frac{DS_p \times g_p}{g_p^j} \tag{11}$$

where DS_p and g_p are the the degree of saturation and green signal time for phase p in the current signal plan, and g_p^j is the degree of saturation for phase p in signal plan j. Then the algorithm selects the signal plan with the optimal DS.

Algorithm 2: Selection of Phase Split in SCATS

Input: Candidate signal plan set $A = \{a_j | j = 1, \ldots, N\}$ and current plan a_c
Output: Signal plan for next cycle
1 (Calculate DS_p for each phase $p \in \{1, \ldots, P\}$ in the current signal plan a_c)
2 **forall** p *in* $\{1, \ldots, P\}$ **do**
3 | Calculate DS_p using Eq. (10) ;
4 **end**
5 (Infer $\bar{DS}_p^j$ for all candidate signal plans)
6 **forall** a_j *in* A **do**
7 | **forall** p *in* $\{1, \ldots, P\}$ **do**
8 | | Calculate $\bar{DS}_p^j$ using Eq. (11) ;
9 | **end**
10 **end**
11 Return the signal plan a_j that minimizes $\Sigma_i^P \bar{DS}_i^j$;

4 REINFORCEMENT LEARNING BASED TRAFFIC SIGNAL CONTROL

Recently, different artificial intelligent techniques have been proposed to control the traffic signal, like fuzzy logic algorithms [Gokulan and Srinivasan 2010], swarm intelligence [Teodorović 2008], and reinforcement learning [El-Tantawy and Abdulhai 2012; Kuyer et al. 2008; Mannion et al. 2016; Wiering 2000]. Among these technologies, RL is more trending these years.

4.1 Preliminaries

4.1.1 Basic concepts. In order to solve the traffic signal control problem using RL, we first introduce the formulation of RL in Problem 1, then we introduce how traffic signal control fits the RL setting. We will also introduce the formulation of multi-agent RL (MARL) in the context of traffic signal control.

Problem 1 (RL Framework). *Usually a single-agent RL problem is modeled as a Markov Decision Process (MDP) $< \mathcal{S}, \mathcal{A}, P, R, \gamma >$, where $\mathcal{S}, \mathcal{A}, P, \mathcal{R}, \gamma$ are the set of state representations, the set of action, the probabilistic state transition function, the reward function, and the discount factor respectively. Given two sets $\mathcal{X}$ and $\mathcal{Y}$, we use $\mathcal{X} \times \mathcal{Y}$ to denote the Cartesian product of $\mathcal{X}$ and $\mathcal{Y}$, i.e., $\mathcal{X} \times \mathcal{Y} = \{(x, y) | x \in \mathcal{X}, y \in \mathcal{Y}\}$. The definitions are given as follows:*

- $\mathcal{S}$: *At time step t, the agent observes state* $\mathrm{s}^t \in \mathcal{S}$.
- $\mathcal{A}, P$: *At time step t, the agent takes an action* $\mathrm{a}^t \in \mathcal{A}$, *which induces a transition in the environment according to the state transition function*

$$P(\mathrm{s}^{t+1}|\mathrm{s}^t, \mathrm{a}^t) : \mathcal{S} \times \mathcal{A} \to \mathcal{S} \tag{12}$$

- R: *At time step t, the agent obtains a reward* r^t *by a reward function.*

$$R(\mathrm{s}^t, \mathrm{a}^t) : \mathcal{S} \times \mathcal{A} \to \mathbb{R} \tag{13}$$

- γ: *The goal of an agent is to find a policy that maximizes the expected return, which is the discounted sum of rewards:*

$$G^t := \sum_{i=0}^{\infty} \gamma^i \mathrm{r}^{t+i} \tag{14}$$

where the discount factor $\gamma \in [0, 1]$ *controls the importance of immediate rewards versus future rewards. Here, we only consider continuing agent-environment intersections which do not end with terminal states but goes on continually without limit.*

Definition 4.1 (Optimal policy and optimal value functions). Solving a reinforcement learning task means, roughly, finding an optimal policy π^* that maximizes expected return. While the agent only receives reward about its immediate, one-step performance, one way to find the optimal policy π^* is by following an optimal *action-value function* or *state-value function*. The action-value function (Q-function) of a policy π, $Q^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, is the expected return of a state-action pair $Q^\pi(\mathrm{s}, \mathrm{a}) = \mathbb{E}_\pi[G^t|\mathrm{s}^t = \mathrm{s}, \mathrm{a}^t = \mathrm{a}]$.

The optimal Q-function is defined as $Q^*(\mathrm{s}, \mathrm{a}) = \max_\pi Q^\pi(\mathrm{s}, \mathrm{a})$. It satisfies the Bellman optimality equation:

$$Q^*(\mathrm{s}^t, \mathrm{a}^t) = \sum_{\mathrm{s}^{t+1} \in \mathcal{S}} P(\mathrm{s}^{t+1}|\mathrm{s}^t, \mathrm{a}^t)[r^t + \gamma \max_{\mathrm{a}^{t+1}} Q^*(\mathrm{s}^{t+1}, \mathrm{a}^{t+1})], \forall \mathrm{s}^t \in \mathcal{S}, \mathrm{a}^t \in \mathcal{A} \tag{15}$$

The state-value function of a policy π, $V^\pi : \mathcal{S} \to \mathbb{R}$, is the expected return of a state $V^\pi(\mathrm{s}) = \mathbb{E}_\pi[G^t|\mathrm{s}^t = \mathrm{s}]$. The optimal state-value function is defined as $V^*(\mathrm{s}) = \max_\pi V^\pi(\mathrm{s})$. It satisfies the Bellman optimality equation:

$$V^*(\mathrm{s}^t) = \max_{\mathrm{a}^t \in \mathcal{A}} \sum_{\mathrm{s}^{t+1} \in \mathcal{S}} P(\mathrm{s}^{t+1}|\mathrm{s}^t, \mathrm{a}^t)[\mathrm{r} + \gamma v_*(\mathrm{s}^{t+1})], \forall \mathrm{s}^t \in \mathcal{S} \tag{16}$$

Example 4.2 (Isolated Traffic Signal Control Problem). Figure 6 illustrates the basic idea of the RL framework in traffic light control problem. The environment is the traffic conditions on the roads, and the agent G controls the traffic signal. At each time step t, a description of the environment (e.g., signal phase, waiting time of cars, queue length of cars, and positions of cars) will be generated as the state s_t. The agent will make a prediction on the next action a^t to take that maximizes the expected return defined as Eq. (14), where the action could be changing to a certain phase in a single intersection scenario. The action a^t will be executed in the environment, and a reward r^t will be generated, where the reward could be defined on the traffic condition of the intersection. Usually, during the decision process, the policy that the agent takes combines the exploitation of learned policy and exploration of a new policy.

Problem 2 (Multiagent RL Framework). *The generalization of the MDP to the multi-agent case is the stochastic game (SG). A stochastic game is defined by a tuple* $\Gamma =< \boldsymbol{S}, \boldsymbol{P}, \boldsymbol{A}, \boldsymbol{R}, \boldsymbol{O}, N, \gamma >$, *where* $\boldsymbol{S}, \boldsymbol{P}, \boldsymbol{A}, \boldsymbol{R}, \boldsymbol{O}, N, \gamma$ *are the sets of states, transition probability functions, joint actions, reward functions, private observations, number of agents and a discount factor respectively. The definitions are given as follows:*

- N: N *agents identified by* $i \in \boldsymbol{I} = \{1, \ldots N\}$.
- $\boldsymbol{S}, \boldsymbol{O}$: *At each time step t, agent i draws observation* $o_i^t \in \boldsymbol{O}$ *correlated with the true environment state* $\mathrm{s}^t \in \boldsymbol{S}$ *according to the observation function* $\boldsymbol{S} \times \boldsymbol{I} \to \boldsymbol{O}$.

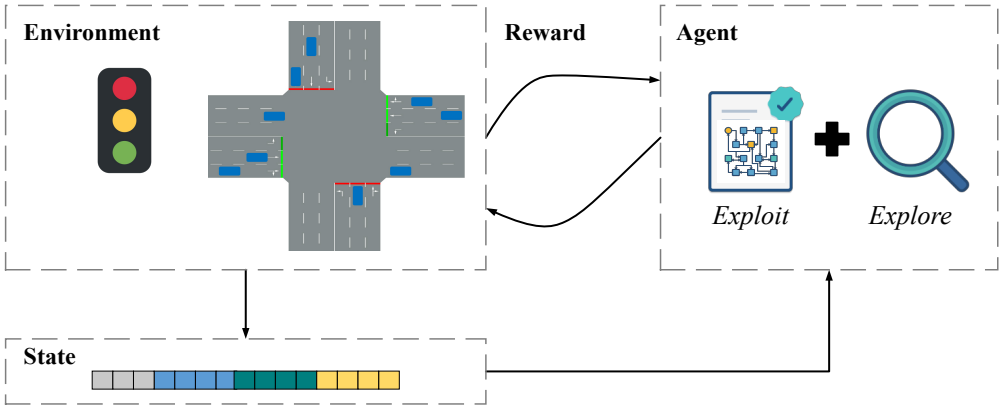

Fig. 6. Reinforcement learning framework for traffic light control.

- $\boldsymbol{P}, \boldsymbol{A}$. *Agent i's action set $\boldsymbol{A}_i$ is defined as a group of phases. At time step t, each agent takes an action $\mathrm{a}_i^t \in \boldsymbol{A}_i$, forming a joint action $\boldsymbol{a}^t = \mathrm{a}_1, \ldots, \mathrm{a}_N$, which induces a transition in the environment according to the state transition function*

$$\boldsymbol{P}(\mathrm{s}^{t+1} | \mathrm{s}^t, \boldsymbol{a}^t) : \boldsymbol{S} \times \boldsymbol{A}_1 \times \cdots \times \boldsymbol{A}_N \to \Omega(\boldsymbol{S}) \tag{17}$$

  where $\Omega(\boldsymbol{S})$ denotes the space of state distributions.
- $\boldsymbol{R}$: *In a stochastic game setting, the reward an agent obtains is also influenced by the actions of other agents. Therefore, at time t, each agent i obtains rewards r_i^t by a reward function*

$$R_i(\mathrm{s}^t, \boldsymbol{a}^t) : \boldsymbol{S} \times \boldsymbol{A}_1 \times \cdots \times \boldsymbol{A}_N \to \mathbb{R} \tag{18}$$

- γ: *Intuitively, the joint actions have long-term effects on the environment. Each agent i chooses an action following a certain policy π_i, aiming to maximize its total reward, $G^t := \sum_{j=0}^{\infty} \gamma^j \mathrm{r}_i^{t+j}$, where the discount factor $\gamma \in [0, 1]$ controls the importance of immediate rewards versus future rewards.*

Example 4.3 (Multi-intersection Traffic Signal Control Problem). For a network with multiple intersections, the agents are defined as the signal controllers for N intersections in the environment. The goal of traffic signal agents controlled with RL is to learn the optimal policies of each agent as well as to optimize the traffic condition of the whole environment. At each timestep t, each agent i observes part of the environment as the observation o_i^t. The agents will make predictions on the next actions $\boldsymbol{a}^t$ to take. In the real world, $\boldsymbol{A}_i$ is mostly pre-defined, i.e., the traffic signal can only change among several phases. The actions will be executed in the environment, and the reward r_i^t will be generated, where the reward could be defined on the level of individual intersections or a group of intersections within the environment.

4.1.2 Basic components of RL for traffic signal control. There are four main components to formulate the problem under the framework of RL:

- Reward design. As RL is learning to maximize a numerical reward, the choice of reward determines the direction of learning.
- State design. State captures the situation on the road and converts it to values. Thus the choice of states should sufficiently describe the environment.
- Selection of action scheme. Different action schemes also have influences on the performance of traffic signal control strategies. For example, if the action of an agent is defined as "which phase to change to", the

traffic signal will have a more flexible phase sequence than defining the action as "keep current phase or change to the next phase in a cycle".

- Coordination strategy. How to achieve coordination is one of the challenges that complicate the signal control problem. In urban environments, signals are often in close proximity, and vehicles departing from one signal influence the arrival pattern of vehicles to the next downstream intersection. Thus, optimizing of signal timings for adjacent traffic signals must be done jointly.

4.2 RL Formulation

A key question for RL is how to formulate the RL setting, i.e., the reward, state and action definition. In recent studies [van der Pol 2016; Wei et al. 2018; Wiering 2000], a typical reward definition for traffic signal control is a weighted linear combination of several components such as queue length, waiting time and delay. The state features include components such as queue length, number of cars, waiting time, and current traffic signal phase. In recent work [van der Pol 2016; Wei et al. 2018], images of vehicles' positions on the roads are also considered in the state and fed into deep neural networks to learn the control policies. For more discussions on the reward, state, and actions, we refer interested readers to [El-Tantawy and Abdulhai 2011; Mannion et al. 2016; Yau et al. 2017].

Table 4. Elements in state definitions

Element	References
Queue length	[Abdoos et al. 2011a, 2014; Aslani et al. 2017, 2018b; Balaji et al. 2010; Brys et al. 2014; Chen et al. 2020; Chin et al. 2011; Chu et al. 2019; El-Tantawy and Abdulhai 2010, 2012; El-Tantawy et al. 2013; Mannion et al. 2016; Nishi et al. 2018; Pham et al. 2013; Prashanth and Bhatnagar 2011; Salkham et al. 2008; Wei et al. 2019b, 2018; Xiong et al. 2019; Xu et al. 2013; Zang et al. 2020; Zheng et al. 2019a,b]
Waiting time	[Chu et al. 2019; Wei et al. 2018]
Volume	[Aslani et al. 2017, 2018b; Balaji et al. 2010; Cahill et al. 2010; Casas 2017; El-Tantawy and Abdulhai 2010; Wei et al. 2019a, 2018]
Delay	[Arel et al. 2010]
Speed	[Casas 2017; El-Tantawy and Abdulhai 2010; Nishi et al. 2018]
Phase duration	[Brys et al. 2014; El-Tantawy et al. 2013; Mannion et al. 2016; Pham et al. 2013; Prashanth and Bhatnagar 2011]
Congestion	[Bakker et al. 2010; Iša et al. 2006; Steingrover et al. 2005]
Position of vehicles	[Bakker et al. 2010; Iša et al. 2006; Khamis and Gomaa 2012; Khamis et al. 2012; Kuyer et al. 2008; Mousavi et al. 2017; Steingrover et al. 2005; van der Pol 2016; Wei et al. 2018; Wiering 2000; Wiering et al. 2004a,b]
Phase	[Aslani et al. 2017, 2018b; Chen et al. 2020; El-Tantawy et al. 2013; Mannion et al. 2016; Salkham et al. 2008; Wei et al. 2019a,b, 2018; Xiong et al. 2019; Zang et al. 2020; Zheng et al. 2019a,b]

4.2.1 State Definitions. At each time step, the agent receives some quantitative descriptions of the environment as the state representation. As is shown in Table 4, various kinds of elements have been proposed to describe the environment state in the traffic signal control problem:

- Queue length. Queue length of a lane is the total number of waiting vehicles on the lane. There are different definitions of a "waiting" state of a vehicle. In [Wei et al. 2018], a vehicle with a speed of less than 0.1 m/s is considered as waiting; in [Bakker et al. 2005; Kuyer et al. 2008], a vehicle without movement in terms of position is considered as waiting.
- Waiting time. The waiting time of a vehicle is defined as the time period a vehicle has been in the "waiting" state (see queue length above for the definition of "waiting"). The definition on the beginning time of a waiting period may be different: in [van der Pol 2016; Wei et al. 2018], they consider the waiting time starting from the last timestamp the vehicle moved, while [Brys et al. 2014; Pham et al. 2013; Wiering 2000] consider the waiting time from the time the vehicle enters the road network.
- Volume. Volume of a lane is defined as the number of vehicles on the lane, which equals to the sum of queuing vehicles and moving vehicles on the lane.
- Delay. Delay of a vehicle is defined as the time a vehicle has traveled within the environment minus the expected travel time (which equals the distance divided by the speed limit).
- Speed. The speed of a vehicle is used to measure how fast the vehicle travels, which could be largely influenced by a pre-defined speed limit. Most literature uses a speed score, which is calculated by vehicle speed divided by the speed limit.
- Phase duration. Phase duration of the current phase is defined as how long the current phase has lasted.
- Congestion. Some studies take the congestion of the outgoing approach into account for effective learning for the cases of congestion and no congestion. The congestion of a lane can be defined either as an indicator (0 for no congestion and 1 for congestion) or the level of congestion, which equals to the number of vehicles divided by the maximum allowed vehicles on the lane.
- Positions of vehicles. The positions of vehicles are usually integrated as an image representation, which is defined as a matrix of vehicle positions, with '1' indicates the presence of a vehicle on a location, and '0' the absence of a vehicle on that location [Mousavi et al. 2017; van der Pol 2016; Wei et al. 2018].
- Phase: The phase information is usually integrated into the state through an index of the current phase in the pre-defined signal phase groups[El-Tantawy et al. 2013; van der Pol 2016; Wei et al. 2018].

There are also some variants of these elements in state representation. These elements can be defined on vehicle level as an image representation with the position of vehicles [van der Pol 2016; Wei et al. 2018]), on lane level by summing or averaging over all vehicles on corresponding lanes [Wei et al. 2018].

Recently, there is a trend of using more complicated states in RL-based traffic signal control algorithms in the hope of gaining a more comprehensive description of the traffic situation. Specifically, recent studies propose to use images [Mousavi et al. 2017; van der Pol 2016; Wei et al. 2018] to represent the state, which results in a state representation with thousands or more dimensions. However, learning with such a high dimension for state often requires a huge number of training samples, meaning that it takes a long time to train the RL agent. And more importantly, longer learning schedule does not necessarily lead to significant performance gain, as the agent may have a more difficult time extracting useful information from the state representation.

4.2.2 Reward Functions. A reward defines the goal in a reinforcement learning problem. Equivalently, we can think of RL as an approach of optimization towards the objective, and this objective is specified as the reward function in the RL context. At each decision point, the agent takes action on the environment, and the environment sends a single numerical value called the reward to the agent. The agent's objective is to maximize the total reward it receives over the long run.

Table 5. Factors in reward functions

Element	References
Queue length	[Abdoos et al. 2011a, 2014; Aslani et al. 2017, 2018b; Balaji et al. 2010; Cahill et al. 2010; Chin et al. 2011; Chu et al. 2019; Iša et al. 2006; Khamis and Gomaa 2012; Khamis et al. 2012; Kuyer et al. 2008; Mannion et al. 2016; Prashanth and Bhatnagar 2011; Salkham et al. 2008; Steingrover et al. 2005; van der Pol 2016; Wei et al. 2019b, 2018; Wiering 2000; Wiering et al. 2004a,b; Xiong et al. 2019; Zang et al. 2020; Zheng et al. 2019a,b]
Waiting time	[Bakker et al. 2010; Brys et al. 2014; Chu et al. 2019; Mannion et al. 2016; Nishi et al. 2018; Pham et al. 2013; Prashanth and Bhatnagar 2011; van der Pol 2016; Wei et al. 2018; Xu et al. 2013]
Change of delay	[Arel et al. 2010; El-Tantawy and Abdulhai 2010, 2012; El-Tantawy et al. 2013; Mousavi et al. 2017]
Speed	[Casas 2017; van der Pol 2016; Wei et al. 2018]
Number of stops	[van der Pol 2016]
Throughput	[Aslani et al. 2017, 2018b; Cahill et al. 2010; Salkham et al. 2008; Wei et al. 2018; Xu et al. 2013]
Frequency of signal change	[van der Pol 2016; Wei et al. 2018]
Accident avoidance	[van der Pol 2016]
Pressure	[Chen et al. 2020; Wei et al. 2019a]

In the traffic signal control problem, although the ultimate objective is to minimize the travel time of all vehicles, travel time is hard to serve as an effective reward in RL for several reasons. First, the travel time of a vehicle is influenced not only by the traffic signals, but also by other factors like the free-flow speed of a vehicle. Second, optimizing the travel time of all vehicles in the network becomes especially harder when the destination of a vehicle in unknown to the traffic signal controller in advance (which is often the case in the real world). Under such circumstances, the travel time of a vehicle can only be measured after it completes its trip when multiple actions have been taken by multiple intersections in the network. Therefore, the reward function is usually defined as a weighted sum of the factors in Table 5 that can be effectively measured after an action:

- Queue length. The queue length is defined as the sum of queue length L over all approaching lanes, where L is calculated as the total number of waiting vehicles on the given lane. Similar to the definition of queue length in Section 4.2.1, there are different definitions on a "waiting" state of a vehicle. Minimizing the queue length is equivalent to minimizing total travel time.
- Waiting time. The waiting time of a vehicle is defined as the time a vehicle has been waiting. Similar to the definition of waiting time in Section 4.2.1, there are different definitions on how to calculate the waiting time of a vehicle. A typical reward function considers the negative value of the waiting time experienced by the vehicles.
- Change of delay. The change (saving) in the total cumulative delay is the difference between the total cumulative delays of two successive decision points. The total cumulative delay at time t is the summation of the cumulative delay, up to time t, of all the vehicles that are currently in the system.

- Speed. A typical reward takes the average speed of all vehicles in the road network. A higher average speed of vehicles in the road network indicates the vehicles travel to their destinations faster.
- Number of stops. A reward can use the average number of stops of all vehicles in the network. Intuitively, the smaller the number of stops, the more smoothly the traffic moves.
- Throughput. The throughput is defined as the total number of vehicles that pass the intersection or leave the network during a certain time interval after the last action. Maximizing the throughput also helps to minimize the total travel time of all vehicles, especially when the road network is congested.
- Frequency of signal change. The frequency of signal change is defined as the number of times the signal changes during a certain time period. Intuitively, the learned policy should not lead to flickering, i.e., changing the traffic signal frequently, as the effective green time for vehicles to pass through the intersection might be reduced.
- Accident avoidance. There are also some studies that have special considerations for accident avoidance. For example, there should not be many emergency stops. Furthermore, jams or would-be collisions should be prevented.
- Pressure of the intersection. In [Wei et al. 2019a], the pressure of an intersection is defined as the sum of absolute pressure of every traffic movement. Intuitively, a higher pressure indicates a higher level of imbalances between the number of incoming lanes and outgoing lanes.

There are also some variants of these factors to measure the immediate reward after an action. The reward could be defined as the values of the factors at certain decision points or defined as the difference between the corresponding total cumulative values over a certain period. Since most of these factors are a result of a sequence of actions and the effect of one action can hardly be reflected, whether to use these factors as the original value or as their difference still remains to be discussed.

Although defining reward as a weighted linear combination of several factors is a common practice in existing studies, there are two concerning issues with these ad-hoc designs in the traffic signal control context. First, there is no guarantee that maximizing the proposed reward is equivalent to optimizing travel time since they are not directly connected in transportation theory. Second, tuning the weights for each reward function component is rather tricky, and minor differences in the weight setting could lead to dramatically different results. Although the factors mentioned in Table 5 are all correlated with travel time, different weighted combinations of them yield very different results. Unfortunately, there is no principled way yet to select those weights.

4.2.3 Action Definitions. Now there are mainly four types of actions as shown in Table 6:

- Set current phase duration. Here, the agent learns to set the duration for the current phase by choosing from pre-defined candidate time periods.
- Set cycle-based phase ratio. Here, the action is defined as the phase split ratio that the signal will set for the next cycle. Usually, the total cycle length is given, and the candidate phase ratio is pre-defined.
- Keep or change the current phase in a cycle-based phase sequence. Here, an action is represented as a binary number, which indicates the agent decides to keep the current phase or change to the next phase.
- Choose the next phase. Decide which phase to change to in a variable phase sequence, in which the phase sequence is not predetermined. Here, the action is the phase index that should be taken next. As a result, this kind of signal timing is more flexible, and the agent is learning to select a phase to change to without assumptions that the signal would change cyclically.

4.3 Learning Approaches

There are varied algorithmic frameworks for RL methods from different perspectives. Depending on whether to learn the state-transition function or not, an RL method can be categorized as a model-based or model-free method,

Table 6. Action definitions

Action	References
Set current phase duration	[Aslani et al. 2017, 2018b; Xu et al. 2013]
Set phase split	[Abdoos et al. 2011a, 2014; Balaji et al. 2010; Casas 2017; Chin et al. 2011]
Keep or change	[Brys et al. 2014; Mannion et al. 2016; Pham et al. 2013; van der Pol 2016; Wei et al. 2018]
Choose next phase	[Arel et al. 2010; Bakker et al. 2010; Cahill et al. 2010; Chen et al. 2020; Chu et al. 2019; El-Tantawy and Abdulhai 2010, 2012; El-Tantawy et al. 2013; Iša et al. 2006; Khamis and Gomaa 2012; Khamis et al. 2012; Kuyer et al. 2008; Mousavi et al. 2017; Nishi et al. 2018; Prashanth and Bhatnagar 2011; Salkham et al. 2008; Steingrover et al. 2005; Wei et al. 2019a,b; Wiering 2000; Wiering et al. 2004a,b; Xiong et al. 2019; Zang et al. 2020; Zheng et al. 2019a,b]

respectively. Depending on whether to learn the value function or to explicitly learn the policy parameter, an RL method can be categorized as a value-based or policy-gradient method, respectively (and the combination of these two is an actor-critic method). Depending on whether the functions, policies, and models are learned through tables with one entry for each state (or state-action pair) or through parameterized function representation, an RL method can be categorized as a tabular or approximation method, respectively.

Table 7. Model-based and model-free methods in RL-based traffic signal control methods

	References	Strengths
Model-based methods	[Cahill et al. 2010; Khamis and Gomaa 2012; Khamis et al. 2012; Kuyer et al. 2008; Steingrover et al. 2005; Wiering 2000; Wiering et al. 2004a,b]	Models the state transitions, explores the state space efficiently by planning and improves convergence speed.
Model-free methods	[Abdoos et al. 2011a, 2014; Arel et al. 2010; Aslani et al. 2017, 2018b; Bakker et al. 2010; Balaji et al. 2010; Brys et al. 2014; Casas 2017; Chen et al. 2020; Chin et al. 2011; Chu et al. 2019; El-Tantawy and Abdulhai 2010, 2012; El-Tantawy et al. 2013; Iša et al. 2006; Mannion et al. 2016; Mousavi et al. 2017; Nishi et al. 2018; Pham et al. 2013; Prashanth and Bhatnagar 2011; Salkham et al. 2008; van der Pol 2016; Wei et al. 2019a,b, 2018; Xiong et al. 2019; Xu et al. 2013; Zang et al. 2020; Zheng et al. 2019a,b]	No need to handcraft/pretrain transition models, learns the models directly with the policy.

4.3.1 Model-based vs. model-free methods. Depending on the modeling philosophy of RL, literature [Arulkumaran et al. 2017; Kaelbling et al. 1996] divides current RL methods into two categories: model-based methods

and model-free methods. For a problem with large state and action space, it is usually difficult to conduct millions of experiments to cover all the possible cases. Model-based methods try to model the transition probability among states explicitly (i.e., learning Equation (12) or (17)), which can be used to sample the environment more efficiently (agents can acquire data samples according to this transition probability). Concretely, given this model, we will know which state each action will take the agent to. In contrast, model-free methods directly estimate the reward for state-action pairs and choose the action based on this. Hence, model-free methods can be used even the transition probability is hard to model.

In the context of traffic signal control, to develop a model-based model, it requires the transition probability of the environment to be known or modeled, like the position, speed, and acceleration of all vehicles, and the operations of all the traffic signals. However, since people's driving behavior is different and hard to predict. Currently, most RL-based methods for traffic signal control are model-free methods, as is shown in Table 7.

4.3.2 Value-based, policy-based vs. actor-critic methods. Depending on the different ways of estimating the potential reward and select action, reinforcement learning methods can be categorized into the following three categories, as is shown in summarized in Table 8:

Table 8. Value- and policy-based methods

	References	Strengths
Value-based	[Abdoos et al. 2011a, 2014; Arel et al. 2010; Bakker et al. 2010; Balaji et al. 2010; Brys et al. 2014; Cahill et al. 2010; Chen et al. 2020; Chin et al. 2011; El-Tantawy and Abdulhai 2010, 2012; El-Tantawy et al. 2013; Iša et al. 2006; Khamis and Gomaa 2012; Khamis et al. 2012; Kuyer et al. 2008; Mannion et al. 2016; Nishi et al. 2018; Pham et al. 2013; Salkham et al. 2008; Steingrover et al. 2005; van der Pol 2016; Wei et al. 2019a,b, 2018; Wiering 2000; Wiering et al. 2004a,b; Xu et al. 2013; Zang et al. 2020; Zheng et al. 2019a,b]	Combines policy evaluation (predicting the value of a policy) and control (finding the best policy), easy to be understood through interpreting the predicted values
Policy-based	[Aslani et al. 2017, 2018b; Casas 2017; Chu et al. 2019; Genders and Razavi 2016; Mousavi et al. 2017; Prashanth and Bhatnagar 2011; Rizzo et al. 2019; Xiong et al. 2019]	Directly learns the policy and optimizes with faster convergence.

- Value-based methods approximate the state-value function or state-action value function (i.e., how rewarding each state is or state-action pair is), and the policy is implicitly obtained from the learned value function. Value-based methods, including Q-learning [Abdoos et al. 2011b; Chin et al. 2011], and DQN [van der Pol 2016; Wei et al. 2018], directly model the state-values or state action values (e.g., under the current traffic situation, how much average speed increase/decrease will take into effect if one action is conducted). In this way, the state and reward can be directly fed into the model without extra processing. However, these methods are usually combined with an ϵ-greedy action selection methods and hence will result in a nearly deterministic policy when ϵ finally decays to a small number (i.e., it is deterministic which action will be conducted under certain states). This may cause the agent to stuck in some unseen or ill-represented cases without improving. In addition, these methods can only deal with discrete actions because it requires a separate modeling process for each action.
- Policy-based methods directly update the policy (e.g., a vector of probabilities to conduct actions under specific state) parameters along the direction to maximize a pre-defined objective (e.g., average expected

reward). Policy-based methods, try to learn a probability distribution of different actions under a certain state. The advantage of policy-based methods is that it does not require the action to be discrete. Besides, it can learn a stochastic policy and keep exploring potentially more rewarding actions. Actor-Critic is one of the widely used methods in policy-based methods. It includes the value-based idea in learning the policy for the action probability distribution, with an actor controls how our agent behaves (policy-based), and the critic measures how good the conducted action is (value-based). Actor-Critic methods in traffic signal control [Aslani et al. 2018a, 2017; Mousavi et al. 2017; Prashanth and Bhatnagar 2011] utilize the strengths of both value function approximation and policy optimization, have shown excellent performance in traffic signal control problems.

4.3.3 Tabular methods and approximation methods. For small-scale discrete reinforcement learning problem in which the state-action pairs can be enumerated, it is a common practice to use a table to record the value functions, policies, or models. However, for a large-scale or continuous problem (state or action is continuous), it is not realistic to enumerate all the state-action pairs. In this case, we need to estimate the value function, policy, or model using an approximation function of the state and action.

Table 9. Tabular methods and approximation methods

	References	Strengths
Tabular methods	[Abdoos et al. 2011a; Balaji et al. 2010; Brys et al. 2014; Cahill et al. 2010; Chin et al. 2011; El-Tantawy and Abdulhai 2010, 2012; El-Tantawy et al. 2013; Iša et al. 2006; Khamis and Gomaa 2012; Khamis et al. 2012; Kuyer et al. 2008; Pham et al. 2013; Prashanth and Bhatnagar 2011; Salkham et al. 2008; Steingrover et al. 2005; Wiering 2000; Wiering et al. 2004a,b; Xu et al. 2013]	Efficient to search the optimal policy
Approximation methods	[Abdoos et al. 2014; Arel et al. 2010; Aslani et al. 2017, 2018b; Bakker et al. 2010; Casas 2017; Chen et al. 2020; Chu et al. 2019; Genders and Razavi 2016; Mannion et al. 2016; Mousavi et al. 2017; Nishi et al. 2018; van der Pol 2016; Wei et al. 2019a,b, 2018; Xiong et al. 2019; Zang et al. 2020; Zheng et al. 2019a,b]	Handles the high dimension of state/action space and generalizes well to unexplored situations.

In the problem scenario of signal control, earlier methods use simple state features like levels of congestion (which can be converted to discrete state values) and employ tabular Q-learning [Abdulhai et al. 2003; El-Tantawy and Abdulhai 2010] to learn the value function. These methods thus, cannot scale up to a large state space for complicated scenarios in traffic signal control. Also, tabular methods will treat samples with similar features as two completely different states, which will decrease the efficiency of utilizing samples in training. To solve these problems, recent studies propose to apply function approximation using continuous state representation[Abdoos et al. 2014; Arel et al. 2010; Arulkumaran et al. 2017; Bazzan and Klügl 2014; Brys et al. 2014; El-Tantawy and Abdulhai 2012; El-Tantawy et al. 2013; Khamis and Gomaa 2012; Mannion et al. 2016; van der Pol 2016; Wei et al. 2018]. The function approximation could be achieved by multiple machine learning techniques, like tile coding [Albus 1971] or deep neural networks. These methods can utilize the samples with similar state values more efficiently and deal with more informative features with continuous ranges, e.g., the position of vehicles.

The categorization of RL-based traffic signal control methods in terms of tabular or approximation is summarized in Table 9.

4.4 Coordination Strategies

Coordination could benefit signal control for multi-intersection scenarios. Since recent advances in RL improve the performance on isolated traffic signal control [van der Pol 2016; Wei et al. 2018], efforts have been performed to design strategies that cooperate with MARL agents. Liturature [Claus and Boutilier 1998] categorizes MARL into two classes: Joint Action Learners and Independent Learners. Here we extend this categorization for the traffic signal control problem, as is shown in Table 10.

- Global single agent. A straightforward solution is to use one central agent to control all the intersections [Prashanth and Bhatnagar 2011]. It directly takes the state as input and learns to set the joint actions of all intersection at the same time. However, this method can result in the curse of dimensionality, which encompasses the exponential growth of the state-action space in the number of state and action dimensions.
- Joint action modeling. [Kuyer et al. 2008] and [van der Pol 2016] consider explicit coordination mechanisms between learning agents using coordination graphs, extending [Wiering 2000] using max-plus algorithm. They factorize the global Q-function as a linear combination of local subproblems: $\hat{Q}(o_1, \ldots, o_N, \boldsymbol{a}) = \Sigma_{i,j} Q_{i,j}(o_i, o_j, \mathrm{a}_i, \mathrm{a}_j)$, where i and j corresponds to the index of neighboring agents.
- Independent RL. There are also a line of studies that use individual RL agents to control the traffic signals in the multi-intersection network [Abdoos et al. 2011a; Balaji et al. 2010; Brys et al. 2014; Cahill et al. 2010; El-Tantawy and Abdulhai 2010; Mannion et al. 2016; Pham et al. 2013; Salkham et al. 2008]. In these methods, each intersection is controlled by an RL agent which senses part of the environment and adapt (or react) to it accordingly and eventually form several subgroups of synchronization.
  - Without communication. These approaches do not use explicit communication to resolve conflicts. Instead, the observation of agent i is defined on the local traffic condition of intersection i. In some simple scenarios like arterial networks, this approach has performed well with the formation of several mini green waves. However, when the environment becomes complicated, the non-stationary impacts from neighboring agents will be brought into the environment, and the learning process usually cannot converge to stationary policies if there are no communication or coordination mechanisms among agents [Nowe et al. 2012]. In highly dynamic environments, an agent might not have the time to perceive a change and adapt to it before the environment has already changed again.
  - With communication. While using individual RL to control each agent in a multi-agent system can form coordination under simple environments, communication between agents can enable agents to behave as a group, rather than a collection of individuals in complex tasks where the environment is dynamic and each agent has limited capabilities and visibility of the world [Sukhbaatar et al. 2016]. It is especially important for traffic signal control in a real-world scenario where the intersections are in close proximity, and the traffic is highly dynamic. While some studies also add neighbor's traffic condition directly into o_i, [Nishi et al. 2018; Wei et al. 2019b] proposes to use Graph Convolutional Network [Schlichtkrull et al. 2018] where cooperating agents learn to communicate amongst themselves. This method not only learns the interactions between the hidden state of neighboring agents and the target agent but also learn multi-hop influences between intersections.

4.5 Experimental Settings

In this section, we will introduce some experimental settings that will influence the performance of traffic signal control strategies: simulation environment, road network setting, and traffic flow setting.

Table 10. Different coordination methods for traffic signal control

Coordination Strategies	Objective & Explanation	References
Global single agent	$max_{\boldsymbol{a}} Q(s, \boldsymbol{a})$, where s is the global environment state, $\boldsymbol{a}$ is the joint action of all intersections	[Casas 2017; Prashanth and Bhatnagar 2011]
Joint action modeling	$max_{a_i,a_j} \Sigma_{i,j} Q_{i,j}(o_i, o_j, \mathrm{a}_i, \mathrm{a}_j)$, where o_i and o_j are the observation of two neighboring agents i and j	[El-Tantawy and Abdulhai 2012; El-Tantawy et al. 2013; Kuyer et al. 2008; van der Pol 2016; Xu et al. 2013]
Independent RL without communication	$max_{\mathrm{a}_i} \Sigma_i Q_i(o_i, \mathrm{a}_i)$, where o_i is the local observation of intersection i, a_i is the action of intersection i	[Abdoos et al. 2011a; Aslani et al. 2017, 2018b; Balaji et al. 2010; Brys et al. 2014; Cahill et al. 2010; Chen et al. 2020; Chu et al. 2019; Iša et al. 2006; Khamis and Gomaa 2012; Khamis et al. 2012; Mannion et al. 2016; Pham et al. 2013; Salkham et al. 2008; Steingrover et al. 2005; Wei et al. 2019a; Wiering 2000; Wiering et al. 2004a,b; Xiong et al. 2019; Zang et al. 2020; Zheng et al. 2019a,b]
Independent RL with communication	$max_{\mathrm{a}_i} \Sigma_i Q_i(\Omega(o_i, \mathcal{N}_i), \mathrm{a}_i)$, where $\mathcal{N}_i$ is the neighborhood representation of intersection i, $\Omega(o_i, \mathcal{N}_i)$ is the function that models local observations and the observations of neighborhoods.	[Arel et al. 2010; El-Tantawy and Abdulhai 2010; Nishi et al. 2018; Wei et al. 2019b; Zhang et al. 2019b]

- Simulation environment. Deploying and testing traffic signal control strategies involves high cost and intensive labor. Hence, simulation is a useful alternative before actual implementation. Simulations of traffic signal control often involve large and heterogeneous scenarios, which should account for some specific mobility models in a vehicular environment, including car following, lane changing, and routing. Since mobility models can significantly affect simulation results, the simulated model must be as close to reality as possible.
- Road network. Different kinds of the road network are explored in the current literature, including synthetic and real-world road network. While most studies conduct experiments on the synthetic grid network, the scale of the network is still relatively small compared to the scale of a city.
- Traffic flow. Traffic flow in the simulation can also influence the performance of control strategies. Usually, the more dynamic and heavier the traffic flow is, the harder for an RL method to learn an optimal policy.

4.5.1 Simulation environment. Various publicly available traffic simulators are currently in use by the research community. In this section, we briefly introduce some open-source tools used in the current traffic signal control literature. Specifically, since RL-based methods require detailed state representation like vehicle-level information,

most literature relies on microscopic simulation, in which movements of individual vehicles are represented through microscopic properties such as the position and velocity of each vehicle. Other proprietary simulators like Paramics [1] or Aimsun [2] will not be introduced here. For a detailed comparison of the open-source simulators, please refer to [Martinez et al. 2011].

Table 11. Different simulators used in literature for RL-based traffic signal control

Name	Latest update	References	Strengths
GLD	1.0 (Jan. 2005)	Strengths [Bakker et al. 2010; Iša et al. 2006; Khamis and Gomaa 2012; Khamis et al. 2012; Kuyer et al. 2008; Prashanth and Bhatnagar 2011; Steingrover et al. 2005; Wiering 2000; Wiering et al. 2004a,b]	Co-learning of vehicle navigation and traffic signal control.
SUMO	1.1.0 (Dec. 2018)	[Chu et al. 2019; Mannion et al. 2016; Mousavi et al. 2017; Nishi et al. 2018; van der Pol 2016; Wei et al. 2018]	Real-time visualization, various vehicle behaviour models, supportive built-on computational frameworks for RL.
AIM	1.0.4 (Mar. 2017)	[Brys et al. 2014; Pham et al. 2013]	Mixed (automated and human-like) vehicle simulation.
CityFlow	0.1 (Mar. 2017)	[Chen et al. 2020; Wei et al. 2019a,b; Xiong et al. 2019; Zang et al. 2020; Zheng et al. 2019a,b]	Multi-thread simulation for large scale traffic settings and multi-agent reinforcement learning, driving behavior modelling.
Others	Paramics [Balaji et al. 2010; El-Tantawy and Abdulhai 2010, 2012; El-Tantawy et al. 2013; Xu et al. 2013], Aimsun [Abdoos et al. 2011a, 2014; Aslani et al. 2017, 2018b; Casas 2017], Matlab [Arel et al. 2010], QLTST [Chin et al. 2011], UTC [Salkham et al. 2008]		

- The Green Light District (GLD) [3]. GLD is an open-source traffic flow simulator that can be used to evaluate the performance of AI traffic light controller algorithms and co-learning navigation algorithms for cars. Vehicles enter the network at edge nodes (i.e., one end of roads which is not connected with an intersection), and each edge node has a certain probability of generating a vehicle at each timestep (spawn rate). Each generated vehicle is assigned to one of the other edge nodes as a destination. The distribution of destinations for each edge node can be adjusted. There can be several types of vehicles, defined by their speed, length, and number of passengers.
- The Autonomous Intersection Management (AIM) [4]. AIM is a microscopic traffic simulator which mainly supports a Manhattan topology of North-South and East-West multi-lane roads joined by many intersections. Developed by the Learning Agents Research Group at the University of Texas at Austin, AIM supports vehicles to navigate, accelerate, and decelerate, as well as subtle details, including variable vehicle sizes.

[1] https://www.paramics-online.com/
[2] https://www.aimsun.com
[3] https://sourceforge.net/projects/stoplicht/
[4] http://www.cs.utexas.edu/ãim/

- Simulation of Urban MObility (SUMO) [5]. SUMO is an open-source program for traffic simulation, mainly developed by the Institute of Transportation Systems, located at German Aerospace Center. Among other features, it allows the existence of different types of vehicles, roads with several lanes, traffic lights, graphical interface to view the network and simulated entities, and interoperability with other applications at run-time through an API called TraCI. Moreover, the tool can be accelerated by allowing a version without a graphical interface. Also, it will enable importing real-world road networks from OpenStreetMap, and the vehicles can enter the system from any position in the network. SUMO also supports other upper-level computational frameworks for deep RL and control experiments for traffic microsimulation.
- CityFlow. CityFlow [Zhang et al. 2019a] is a multi-agent reinforcement learning environment for large scale city traffic scenarios. It supports real-world definitions for the road network and traffic flow and is capable of multi-thread simulation for city-wide traffic settings. It also provides APIs for reinforcement learning, which is suitable for tasks like traffic signal control and driving behavior modeling.

Table 12. Different road network in literature for RL-based traffic signal control

	Intersections	References
Synthetic network	< 10	[Abdoos et al. 2014; Mannion et al. 2016; Xu et al. 2013] [Brys et al. 2014; Nishi et al. 2018; Pham et al. 2013; Wiering 2000] [Arel et al. 2010; Prashanth and Bhatnagar 2011; van der Pol 2016]
	$10 - 20$	[Bakker et al. 2010; Cahill et al. 2010; Iša et al. 2006; Kuyer et al. 2008] [Khamis and Gomaa 2012; Steingrover et al. 2005; Wiering et al. 2004a,b] [Khamis et al. 2012]
	>= 20	[Chu et al. 2019](25 intersections),[Abdoos et al. 2011a] (50 intersections) [Wei et al. 2019b](100)
Real road network		[Balaji et al. 2010](29 intersections), [Casas 2017](43 intersections), [Aslani et al. 2017, 2018b](50 intersections), [Salkham et al. 2008](64 intersections), [El-Tantawy and Abdulhai 2012; El-Tantawy et al. 2013](59 intersections), [Zheng et al. 2019b](16 intersections), [Zheng et al. 2019a](5 intersections) [Wei et al. 2019a](16 intersections), [Wei et al. 2019b](196 intersections), [Chen et al. 2020] (2150 intersections)

4.5.2 Road network. At a coarse scale, a road network is a directed graph with nodes and edges representing intersections and roads, respectively. Specifically, a real-world road network can be more complicated than the synthetic network in the road properties (like the position, shape, and speed limit of every lane), intersections, traffic movements, traffic signal logic (movement signal phases). Table 12 summarizes the road network in literature. While existing studies conduct experiments on different kinds of road networks, the number of intersections in the tested network is mostly smaller than twenty.

4.5.3 Traffic flow. In the traffic flow dataset, each vehicle is described as (o, t, d), where o is the origin location, t is time, and d is the destination location. Locations o and d are both locations on the road network. Table 13 summarizes the traffic flow utilized in the current studies. In the synthetic traffic flow data, either o, t or d could be synthesized to generate uniform or dynamic changing flow in different levels of traffic.

[5] http://sumo.sourceforge.net

Table 13. Traffic flow used in literature for RL-based traffic signal control. Traffic with arrival rate less than 500 vehicles/hour/lane is considered as light traffic in this survey, otherwise considered as heavy.

			References
Synthetic data	light	uniform	[Mousavi et al. 2017; Wei et al. 2018; Wiering 2000; Xu et al. 2013]
		dynamic	[Abdoos et al. 2011a, 2014; Chin et al. 2011; Mannion et al. 2016] [Arel et al. 2010; Bakker et al. 2010; Kuyer et al. 2008; Salkham et al. 2008] [Nishi et al. 2018; van der Pol 2016]
	heavy	uniform	[Brys et al. 2014; Pham et al. 2013; Prashanth and Bhatnagar 2011] [Steingrover et al. 2005; Wei et al. 2018; Wiering et al. 2004a,b]
		dynamic	[Abdoos et al. 2011a; Arel et al. 2010; Balaji et al. 2010; Wei et al. 2018] [Bakker et al. 2010; Khamis and Gomaa 2012; Khamis et al. 2012] [Chu et al. 2019; Steingrover et al. 2005; Zheng et al. 2019a,b]
Real data			[Casas 2017; El-Tantawy and Abdulhai 2010, 2012; El-Tantawy et al. 2013; Wei et al. 2018] [Aslani et al. 2017, 2018b; Zheng et al. 2019a,b] [Chen et al. 2020; Wei et al. 2019a,b; Xiong et al. 2019; Zang et al. 2020]

4.6 Challenges in RL for Traffic Signal Control

Current RL-based methods have the following challenges:

4.6.1 Design of RL formulation. A key question for RL is how to formulate the RL setting, i.e., the reward and state definition. In existing studies [van der Pol 2016; Wei et al. 2018; Wiering 2000], a typical reward definition for traffic signal control is a weighted linear combination of several components such as queue length, waiting time and delay. The state features include components such as queue length, number of cars, waiting time, and current traffic signal phase. In recent work [van der Pol 2016; Wei et al. 2018], images of vehicles' positions on the roads are also considered in the state.

However, all of the existing work take an ad-hoc approach to define reward and state. Such an ad-hoc approach will cause several problems that hinder the application of RL in the real world. First, the engineering details in formulating the reward function and state feature could significantly affect the results. For example, if the reward is defined as a weighted linear combination of several terms, the weight on each term is tricky to set, and a minor difference in weight setting could lead to dramatically different results. Second, the state representation could be in a high-dimensional space, especially when using traffic images as part of the state representation [van der Pol 2016; Wei et al. 2018]. With such a high-dimensional state representation, the neural network will need much more training data samples to learn and may not even converge. Third, there is no connection between existing RL approaches and transportation methods. Without the support of transportation theory, it is highly risky to apply these purely data-driven RL-based approaches in the real physical world.

Learning efficiency. While learning from trial-and-error is the key idea in RL, the learning cost of RL could be unacceptable for complicated problems. Existing RL methods for games (e.g., Go or Atari games) usually require a massive number of update iterations of RL models to yield impressive results in simulated environments. These trial-and-error attempts will lead to real traffic jams in the traffic signal control problem. Therefore, how to learn efficiently (e.g., learning from limited data samples, efficient exploration, transferring learned knowledge) is a critical question for the application of RL in traffic signal control.

Credit assignment. The credit assignment problem is one of the heatedly investigated problems in RL, which considers the distribution of credits for success (or blame for failure) of an action [Sutton 1984]. In the traffic signal control problem, the traffic condition is a consequence of several actions that traffic signal controllers have been taken, which brings two problems: (1) one action may still have effect after several steps of actions; (2) the reward at each timestamp is a result of the combination of consequent actions from several agents. Unlike in Atari or Go games, where people sometimes assign the final score of an episode to all the actions associated with this episode, the action in the traffic signal control problem may have an affecting time interval that may be dynamically changing and needs to be further investigated.

Safety issue. Making reinforcement learning agents acceptably safe in physical environments is another important area for future research. While RL methods learn from trial-and-error, the learning cost of RL could be critical or even fatal in the real world as the malfunction of traffic signals might lead to accidents. Therefore, how to adopt risk management into RL helps prevent unwanted behavior during and after the learning process of RL.

5 CONCLUSION

In this article, we present an overview of traffic signal control. We first introduce some terms and objectives in traffic signal control problem. Then we introduce some of the classical transportation approaches. Next we review state-of-the-art RL-based traffic signal control methods from the perspective of the formulation of RL agent (state, reward, action and ways of coordination) and their experiment settings. After that, we briefly discuss some challenges for future research directions on RL-based traffic signal control methods. Hopefully, this survey can provide a comprehensive view from both traditional transportation methods and reinforcement learning methods and can bring the traffic signal control research into a new frontier.

A APPENDIX A

A.1 Summary of RL methods

RL-based traffic signal control methods are summarised in the Table 14.

Table 14. Overall comparison of RL-based traffic signal control methods investigated in this survey

Paper name	State	Reward	Action	Model	Method	Approx	Coope-ration	Simula-tor	Road net	Traffic flow
[Abdoos et al. 2011a]	1	1	1	2	1	1	2	6	3(50)	2,4
[Abdoos et al. 2014]	1	1	1	2	1	2	1	6	1(9)	2
[Balaji et al. 2010]	1,3	1	1	2	1	1	2	5	4(29)	4
[Cahill et al. 2010]	3	1,7	3	1	1	1	2	1	2(15)	uk
[Chin et al. 2011]	1	1	1	2	1	1	-	7	5	2
[El-Tantawy and Abdulhai 2010]	1,3,4	3	3	2	1	1	3	5	5	5
[El-Tantawy and Abdulhai 2012]	1	3	3	2	1	1	4	5	4(59)	5
[El-Tantawy et al. 2013]	1,6,9	3	3	2	1	1	4	5	4(59)	5
[Mannion et al. 2016]	1,6,9	1,2	2	2	1	2	2	2	1(9)	2
[Salkham et al. 2008]	1,9	1,7	3	2	1	1	2	8	4(64)	2
[Wei et al. 2018]	1,2,3,8,9	1,2,5,7,8	2	2	1	2	-	2	5	1,3,4,5
[Xu et al. 2013]	1	2,7	4	2	1	1	4	4	1(5)	1
[Brys et al. 2014]	1,6	2	2	2	1	1	2	3	1(4)	3
[Pham et al. 2013]	1,6	2	2	2	1	1	2	3	1(4)	3
[Wiering 2000]	8	2	3	1	1	1	2	1	1(6)	1
[Prashanth and Bhatnagar 2011]	1,6	1,2	3	2	2	1	1	1	1(5)	3
[Arel et al. 2010]	5	3	3	2	1	2	3	4	1(5)	2,4
[Bakker et al. 2010]	7,8	2	3	2	1	2	2,4	1	2(15)	2,4
[Iša et al. 2006]	7,8,10	2	3	2	1	1	2	1	2(15)	uk
[Kuyer et al. 2008]	8	2	3	1	1	1	4	1	2(15)	2
[Wiering et al. 2004a]	8	2	3	1	1	1	2	1	2(15)	3
[Wiering et al. 2004b]	8	2	3	1	1	1	2	1	2(15)	3
[van der Pol 2016]	8	1,2,5,6,8	2	2	1	2	4	2	1(4)	2
[Khamis and Gomaa 2012]	8	1	3	1	1	1	2	1	2(12)	4
[Khamis et al. 2012]	8	1	3	1	1	1	2	1	2(12)	4
[Mousavi et al. 2017]	8	3	3	2	2	2	-	2	5	1
[Casas 2017]	3,4	5	1	2	2	2	1	6	4(43)	5
[Aslani et al. 2017]	1,3,9	1,7	4	2	2	2	2	6	4(50)	5
[Aslani et al. 2018b]	1,3,9	1,7	4	2	2	2	2	6	4(50)	5
[Nishi et al. 2018]	1,4	2	3	2	1	2	3	2	1(6)	2
[Chu et al. 2019]	1,2	1,2	3	2	2	2	2	2	3(25)	4
[Zheng et al. 2019b]	1,9	1	3	2	1	2	2	2	4(16)	4
[Zheng et al. 2019a]	1,9	1	3	2	1	2	2	2	4(5)	4
[Wei et al. 2019a]	3,9	9	3	2	1	2	2	9	4(16)	5
[Wei et al. 2019b]	1,9	1	3	2	1	2	3	2	4(48)	5

Table 15. Overall comparison of RL-based traffic signal control methods investigated in this survey

Paper name	State	Reward	Action	Model	Method	Approx	Coope-ration	Simula-tor	Road net	Traffic flow
[Zang et al. 2020]	1,9	1	3	2	1	2	-	9	4(16)	4
[Xiong et al. 2019]	1,9	1	3	2	1	2	-	9	4(5)	4
[Chen et al. 2020]	3,9	9	3	2	1	2	2	9	4(16)	5
[Rizzo et al. 2019]	2	7	3	2	1	2	3	2	4(1)	5
[Wang et al. 2019]	1,2	1	2	2	1	2	3	4	4(4)	5
[Zhang et al. 2019b]	1,2,3	1,2,5,7,8	2	2	1	2	3	2	3(36)	4

State: 1. Queue; 2. Waiting time; 3. Volume; 4. Speed; 5. Delay; 6. Elapsed time; 7. Congestion; 8. Position of vehicles; 9. Phase; 10. Accident; 11. Pressure
Reward: 1.Queue; 2. Waiting time; 3.Delay; 4. Accident; 5. Speed; 6. Number of stops; 7. Throughput 8. Frequency of signal change; 9. Pressure
Action: 1. Phase split; 2. phase switch; 3. phase itself; 4 phase duration.
Model: 1. Model-based; 2.Model-free.
Method: 1. Value-based; 2. Policy-based
Approx: 1. Tab; 2. Approx.
Cooperation: 1. Single; 2. IRL wo/ communication; 3. IRL w/ communication; 4. Joint action.
Simulation: 1. GLD; 2. SUMO; 3. AIM; 4. Matlab; 5. Paramics; 6. Aimsun; 7. QLTST; 8. UTC 9. CityFlow
Road net: 1. Synthetic <10; 2. Synthetic 11-20; 3. Synthetic >=21; 4. Real 5. Single intersection
Traffic flow: 1. Synthetic light uniform; 2. Synthetic light dynamic; 3. Synthetic heavy uniform; 4. Synthetic heavy dynamic; 5. Real-world data

ACKNOWLEDGMENTS

This is acknowledgement.

REFERENCES

Monireh Abdoos, Nasser Mozayani, and Ana LC Bazzan. 2011a. Traffic light control in non-stationary environments based on multi agent Q-learning. In *ITSC*. IEEE.

Monireh Abdoos, Nasser Mozayani, and Ana LC Bazzan. 2011b. Traffic light control in non-stationary environments based on multi agent Q-learning. In *ITSC*. IEEE.

Monireh Abdoos, Nasser Mozayani, and Ana LC Bazzan. 2014. Hierarchical control of traffic signals using Q-learning with tile coding. *Applied intelligence* 40, 2 (2014), 201–213.

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.

James S Albus. 1971. A theory of cerebellar function. *Mathematical Biosciences* 10, 1-2 (1971), 25–61.

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.

Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. 2017. A brief survey of deep reinforcement learning. *arXiv preprint* (2017).

Mohammad Aslani, Mohammad Saadi Mesgari, Stefan Seipel, and Marco Wiering. 2018a. Developing adaptive traffic signal control by actor–critic and direct exploration methods. In *Proceedings of the Institution of Civil Engineers-Transport*. Thomas Telford Ltd, 1–10.

Mohammad Aslani, Mohammad Saadi Mesgari, and Marco Wiering. 2017. Adaptive traffic signal control with actor-critic methods in a real-world traffic network with different traffic disruption events. *Transportation Research Part C: Emerging Technologies* 85 (2017), 732–752.

Mohammad Aslani, Stefan Seipel, Mohammad Saadi Mesgari, and Marco Wiering. 2018b. Traffic signal optimization through discrete and continuous reinforcement learning with robustness analysis in downtown Tehran. *Advanced Engineering Informatics* 38 (2018), 639–655.

Bram Bakker, M Steingrover, Roelant Schouten, EHJ Nijhuis, LJHM Kester, et al. 2005. Cooperative multi-agent reinforcement learning of traffic lights. (2005).

Bram Bakker, Shimon Whiteson, Leon Kester, and Frans CA Groen. 2010. Traffic light control by multiagent reinforcement learning systems. In *Interactive Collaborative Information Systems*. Springer.

PG Balaji, X German, and Dipti Srinivasan. 2010. Urban traffic signal control using reinforcement learning agents. *IET Intelligent Transport Systems* 4, 3 (2010), 177–188.

Ana LC Bazzan and Franziska Klügl. 2014. A review on agent-based technology for traffic and transportation. *The Knowledge Engineering Review* 29, 3 (2014), 375–403.

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.

Vinny Cahill et al. 2010. Soilse: A decentralized approach to optimization of fluctuating urban traffic using reinforcement learning. In *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*. IEEE, 531–538.

Noe Casas. 2017. Deep deterministic policy gradient for urban traffic light control. *arXiv preprint* (2017).

Chacha Chen, Hua Wei, Nan Xu, Guanjie Zheng, et al. 2020. Toward A Thousand Lights: Decentralized Deep Reinforcement Learning for Large-Scale Traffic Signal Control. In *AAAI*.

Yit Kwong Chin, N Bolong, Soo Siang Yang, and KTK Teo. 2011. Exploring Q-learning optimization in traffic signal timing plan management. In *Computational Intelligence, Communication Systems and Networks (CICSyN), 2011 Third International Conference on*. IEEE, 269–274.

Tianshu Chu, Jie Wang, Lara Codecà, and Zhaojian Li. 2019. Multi-Agent Deep Reinforcement Learning for Large-scale Traffic Signal Control. *arXiv preprint* (2019).

Caroline Claus and Craig Boutilier. 1998. The dynamics of reinforcement learning in cooperative multiagent systems. *AAAI/IAAI* 1998 (1998), 746–752.

CNN. 2019. Alibaba's 'City Brain' is slashing congestion in its hometown. https://edition.cnn.com/2019/01/15/tech/alibaba-city-brain-hangzhou/index.html. (January 2019).

Seung-Bae Cools, Carlos Gershenson, and Bart DâĂŹHooghe. 2013. Self-organizing traffic lights: A realistic simulation. In *Advances in applied self-organizing systems*. Springer, 45–55.

The Economist. 2014. The cost of traffic jams. https://www.economist.com/blogs/economist-explains/2014/11/economist-explains-1. (November 2014).

Samah El-Tantawy and Baher Abdulhai. 2010. An agent-based learning towards decentralized and coordinated traffic signal control. In *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*. IEEE, 665–670.

Samah El-Tantawy and Baher Abdulhai. 2011. *Comprehensive analysis of reinforcement learning methods and parameters for adaptive traffic signal control*. Technical Report.

Samah El-Tantawy and Baher Abdulhai. 2012. Multi-agent reinforcement learning for integrated network of adaptive traffic signal controllers (MARLIN-ATSC). In *ITSC*. IEEE, 319–326.

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.

Nathan H Gartner, Susan F Assman, Fernando Lasaga, and Dennis L Hou. 1991. A multi-band approach to arterial traffic signal optimization. *Transportation Research Part B: Methodological* 25, 1 (1991), 55–74.

Wade Genders and Saiedeh Razavi. 2016. Using a deep reinforcement learning agent for traffic signal control. *arXiv preprint* (2016).

Carlos Gershenson. 2004. Self-organizing traffic lights. *arXiv preprint* (2004).

Balaji Parasumanna Gokulan and Dipti Srinivasan. 2010. Distributed geometric fuzzy multiagent urban traffic signal control. *IEEE Transactions on Intelligent Transportation Systems* 11, 3 (2010), 714–727.

PB Hunt, DI Robertson, RD Bretherton, and M Cr Royle. 1982. The SCOOT on-line traffic signal optimisation technique. *Traffic Engineering & Control* 23, 4 (1982).

PB Hunt, DI Robertson, RD Bretherton, and RI Winton. 1981. *SCOOT-a traffic responsive method of coordinating signals*. Technical Report.

Jirı Iša, Julian Kooij, Rogier Koppejan, and Lior Kuijer. 2006. Reinforcement learning of traffic light controllers adapting to accidents. *Design and Organisation of Autonomous Systems* (2006), 1–14.

Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. 1996. Reinforcement learning: A survey. *Journal of artificial intelligence research* 4 (1996), 237–285.

Mohamed Abdelaziz Khamis and Walid Gomaa. 2012. Enhanced multiagent multi-objective reinforcement learning for urban traffic light control. In *Machine Learning and Applications (ICMLA)*, Vol. 1. IEEE, 586–591.

Mohamed A Khamis, Walid Gomaa, and Hisham El-Shishiny. 2012. Multi-objective traffic light control system based on Bayesian probability interpretation. In *ITSC*. IEEE.

Peter Koonce et al. 2008. *Traffic signal timing manual*. Technical Report. United States. Federal Highway Administration.

Lior Kuyer, Shimon Whiteson, Bram Bakker, and Nikos Vlassis. 2008. Multiagent reinforcement learning for urban traffic control using coordination graphs. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 656–671.

Li Li, Ding Wen, and Danya Yao. 2014. A survey of traffic control with vehicular communications. *IEEE TITS* 15, 1 (2014).

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

P Lowrie. 1990. SCATS–A Traffic Responsive Method of Controlling Urban Traffic. Roads and Traffic Authority, Sydney. *New South Wales, Australia* (1990).

PR Lowrie. 1992. SCATS–a traffic responsive method of controlling urban traffic. Roads and traffic authority. *NSW, Australia* (1992).

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*.

Francisco J Martinez, Chai Keong Toh, Juan-Carlos Cano, Carlos T Calafate, and Pietro Manzoni. 2011. A survey and comparative study of simulators for vehicular ad hoc networks (VANETs). *Wireless Communications and Mobile Computing* (2011).

Seyed Sajad Mousavi, Michael Schukat, and Enda Howley. 2017. Traffic light control using deep policy-gradient and value-function-based reinforcement learning. *Intelligent Transport Systems* (2017).

Tomoki Nishi, Keisuke Otaki, Keiichiro Hayakawa, and Takayoshi Yoshimura. 2018. Traffic Signal Control Based on Reinforcement Learning with Graph Convolutional Neural Nets. In *ITSC*. IEEE, 877–883.

Ann Nowe, Peter Vrancx, and Yann MichaÃńl De Hauwere. 2012. *Game Theory and Multi-agent Reinforcement Learning*.

Markos Papageorgiou, Christina Diakaki, Vaya Dinopoulou, Apostolos Kotsialos, and Yibing Wang. 2003. Review of road traffic control strategies. *Proc. IEEE* 91, 12 (2003), 2043–2067.

Tong Thanh Pham, Tim Brys, Matthew E Taylor, Tim Brys, et al. 2013. Learning coordinated traffic light control. In *AAMAS*.

L A Prashanth and Shalabh Bhatnagar. 2011. Reinforcement learning with average cost for adaptive control of traffic lights at intersections. *ITSC* (2011).

Stefano Giovanni Rizzo, Giovanna Vantini, and Sanjay Chawla. 2019. Time Critic Policy Gradient Methods for Traffic Signal Control in Complex and Congested Scenarios. In *KDD*.

Roger P Roess, Elena S Prassas, and William R McShane. 2004. *Traffic engineering*. Pearson.

As' ad Salkham, Raymond Cunningham, Anurag Garg, and Vinny Cahill. 2008. A collaborative reinforcement learning approach to urban traffic control optimization. In *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology-Volume 02*. IEEE Computer Society, 560–566.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*. Springer, 593–607.

David Schrank, Bill Eisele, and Tim Lomax. 2012. TTIâĂŹs 2012 urban mobility report. *Texas A&M Transportation Institute. The Texas A&M University System* 4 (2012).

David Schrank, Bill Eisele, Tim Lomax, and Jim Bak. 2015. 2015 Urban Mobility Scorecard. (2015).

Chronis Stamatiadis and Nathan Gartner. 1996. MULTIBAND-96: a program for variable-bandwidth progression optimization of multiarterial traffic networks. *Transportation Research Record: Journal of the Transportation Research Board* 1554 (1996), 9–17.

Merlijn Steingrover, Roelant Schouten, Stefan Peelen, Emil Nijhuis, Bram Bakker, et al. 2005. Reinforcement Learning of Traffic Light Controllers Adapting to Traffic Congestion.. In *BNAIC*. Citeseer, 216–223.

Aleksandar Stevanovic. 2010. *Adaptive traffic control systems: domestic and foreign state of practice*. Number Project 20-5 (Topic 40-03).

Sainbayar Sukhbaatar, Rob Fergus, et al. 2016. Learning multiagent communication with backpropagation. In *NeurIPS*. 2244–2252.

Richard Stuart Sutton. 1984. Temporal credit assignment in reinforcement learning. (1984).

Dušan Teodorović. 2008. Swarm intelligence systems for transportation engineering: Principles and applications. *Transportation Research Part C: Emerging Technologies* 16, 6 (2008), 651–667.

Elise van der Pol. 2016. Coordinated Deep Reinforcement Learners for Traffic Light Control. NeurIPS.

Pravin Varaiya. 2013. The max-pressure controller for arbitrary networks of signalized intersections. In *Advances in Dynamic Network Modeling in Complex Transportation Systems*. Springer, 27–66.

Yanan Wang, Tong Xu, Xin Niu, Chang Tan, Enhong Chen, and Hui Xiong. 2019. STMARL: A Spatio-Temporal Multi-Agent Reinforcement Learning Approach for Traffic Light Control. *arXiv preprint* (2019).

Hua Wei, Chacha Chen, Guanjie Zheng, Kan Wu, Vikash Gayah, Kai Xu, and Zhenhui Li. 2019a. PressLight: Learning Max Pressure Control to Coordinate Traffic Signals in Arterial Network. In *KDD*.

Hua Wei, Nan Xu, Huichu Zhang, Guanjie Zheng, Xinshi Zang, Chacha Chen, Weinan Zhang, Yanmin Zhu, Kai Xu, and Zhenhui Li. 2019b. CoLight: Learning Network-level Cooperation for Traffic Signal Control. In *CIKM*.

Hua Wei, Guanjie Zheng, Huaxiu Yao, and Zhenhui Li. 2018. IntelliLight: A Reinforcement Learning Approach for Intelligent Traffic Light Control. In *KDD*.

MA Wiering. 2000. Multi-agent reinforcement learning for traffic light control. In *ICML*.

MA Wiering, J van Veenen, Jilles Vreeken, and Arne Koopman. 2004a. Intelligent traffic light control. (2004).

Marco Wiering, Jilles Vreeken, Jelle Van Veenen, and Arne Koopman. 2004b. Simulation and optimization of traffic in a city. In *Intelligent Vehicles Symposium, 2004 IEEE*. IEEE, 453–458.

Business Wire. 2018. DiDi's Introduces Smart Transportation Solution for Traffic Management. https://www.businesswire.com/news/home/20180125006404/en/DiDi-Introduces-Smart-Transportation-Solution-Traffic-Management. (January 2018). Accessed: 2019-02-19.

Yuanhao Xiong, Guanjie Zheng, Kai Xu, and Zhenhui Li. 2019. Learning Traffic Signal Control from Demonstrations. In *CIKM*.

Lun-Hui Xu, Xin-Hai Xia, and Qiang Luo. 2013. The study of reinforcement learning for traffic self-adaptive control under multiagent markov game environment. *Mathematical Problems in Engineering* 2013 (2013).

Kok-Lim Alvin Yau, Junaid Qadir, Hooi Ling Khoo, et al. 2017. A Survey on Reinforcement Learning Models and Algorithms for Traffic Signal Control. *ACM Computing Survey* (2017).

Xinshi Zang, Huaxiu Yao, Guanjie Zheng, Nan Xu, Kai Xu, and Zhenhui Li. 2020. MetaLight: Value-based Meta-reinforcement Learning for Online Universal Traffic Signal Control. In *AAAI*.

Huichu Zhang, Siyuan Feng, Chang Liu, et al. 2019a. CityFlow: A Multi-Agent Reinforcement Learning Environment for Large Scale City Traffic Scenario. In *The WebConf*.

Zhi Zhang, Jiachen Yang, and Hongyuan Zha. 2019b. Integrating independent and centralized multi-agent reinforcement learning for traffic signal network optimization. *arXiv preprint* (2019).

Guanjie Zheng, Yuanhao Xiong, Xinshi Zang, Jie Feng, Hua Wei, et al. 2019a. Learning Phase Competition for Traffic Signal Control. In *CIKM*.

Guanjie Zheng, Xinshi Zang, Nan Xu, Hua Wei, Zhengyao Yu, Vikash Gayah, Kai Xu, and Zhenhui Li. 2019b. Diagnosing Reinforcement Learning for Traffic Signal Control. *arXiv preprint* (2019).