Multiple intersections traffic signal control based on cooperative multi-agent reinforcement learning

Junxiu Liu [a], Sheng Qin [a], Min Su [a,*], Yuling Luo [a,*], Yanhu Wang [a], Su Yang [b]

[a] *Guangxi Key Lab of Brain-inspired Computing and Intelligent Chips, School of Electronic and Information Engineering, Guangxi Normal University, China*

[b] *Department of Computer Science, Swansea University, Swansea, UK*

ARTICLE INFO

Keywords:
Traffic signal control
Reinforcement learning
Multi-agent system

ABSTRACT

For the multi-agent traffic signal controls, the traffic signal at each intersection is controlled by an independent agent. Since the control policy for each agent is dynamic, when the traffic scale is large, the adjustment of the agent's policy brings non-stationary effects over surrounding intersections, leading to the instability of the overall system. Therefore, there is the necessity to eliminate this non-stationarity effect to stabilize the multi-agent system. A collaborative multi- agent reinforcement learning method is proposed in this work to enable the system to overcome the instability problem through a collaborative mechanism. Decentralized learning with limited communication is used to reduce the communication latency between agents. The Shapley value reward function is applied to comprehensively calculate the contribution of each agent to avoid the influence of reward function coefficient variation, thereby reducing unstable factors. The Kullback-Leibler divergence is then used to distinguish the current and historical policies, and the loss function is optimized to eliminate the environmental non-stationarity. Experimental results demonstrate that the average travel time and its standard deviation are reduced by using the Shapley value reward function and optimized loss function, respectively, and this work provides an alternative for traffic signal controls on multiple intersections.

1. Introduction

Adaptive Traffic Signal Control (ATSC) is an effective way to reduce traffic congestion (i.e., reducing the average travel time of vehicles). The genetic algorithm [3], swarm intelligence [28,9], Reinforcement Learning (RL) [41] are applied to the ATSC field in previous studies. RL is a promising adaptive decision-making method because it does not require any additional assumptions about the transition and distribution of the controlled system [5]. Unfortunately, troubled by the curse of dimensionality, the performance of RL is greatly reduced. In recent years, with the great development of deep learning [16,15,13,14,17], the combination of RL and deep learning can effectively solve this defect, which is called deep RL (DRL). Therefore, DRL is increasingly used in ATSC research [5,35,20,33].

Generally, the application of DRL for ATSC focuses on two settings: independent-agent setting and multi-agent setting. Independent Q-learning [31] is a commonly used independent-agent model, in which each intersection is controlled by an independent

* Corresponding authors.
E-mail addresses: j.liu@ieee.org (J. Liu), qs_qinsheng@163.com (S. Qin), sumin0303@gxnu.edu.cn (M. Su), yuling0616@gxnu.edu.cn (Y. Luo), 1695017072@qq.com (Y. Wang), su.yang@swansea.ac.uk (S. Yang).

Q-learning agent. This model possesses the advantage of being completely scalable. However, due to the fact that the agent has only partial observability and with a lack of cooperation it cannot get global information about the traffic environment. This makes agents rely only on limited information while making decisions. Thus, it is difficult to obtain the global optimal strategy in large-scale traffic scenarios. In the ATSC, neighbouring intersections interact with each other which increases the adaptability comparing to independent-agent settings. On the contrary, in a multi-agent setting, agents learn and collaborate through mutual communication, and obtain more complete environment information. Centralized critic [26] and parameters sharing [4] are two useful Multi-Agent RL (MARL) methods. The former has a shared critic model and multiple actors. The centralized critic model collects the experience of every actor to train the neural network, which makes the critic model understand the knowledge of the environment to eliminate non-stationary and partial observability. The latter uses the same model on each intersection by sharing parameters. However, both methods need global information sharing, and communication latency will cause them to fail [5]. Decentralized training with limited communication is a useful method for reducing communication latency [5]. In addition to communicating with each other, constructing the global reward of a MARL agent through the local reward of itself and neighbours is also an effective way to encourage cooperation between agents [5]. Linear reward functions are always susceptible to changes in its coefficients [34]. In other words, the difference in coefficients of the reward function will lead to different results, and even these results vary greatly. Therefore, it is very important to design a reasonable and effective reward function.

Besides, in order to obtain a better decentralized training effect, each agent is always adjusting its policy. This leads to changes in the perceived transition and rewards of each agent, which is the non-stationary problem [26]. Furthermore, experience replay [22] is the key technology behind many recent advances in DRL [18]. Unfortunately, the MARL system may become unstable when the experience replay technology is adopted in a non-stationary environment [26]. Due to the continuous adjustment of agent policy, some previous experiences in the experience buffer may be outdated, which will cause the agent to misunderstand the behaviour of other agents and even incorrectly estimate the transform of the environment. Therefore, it is necessary to deal with outdated experiences.

In summary, the problems in a cooperation MARL approach that need to be solved can be summarized as follows: (1) The communication latency caused by global information sharing is high; (2) The reward function used to encourage cooperation between agents is susceptible to coefficients; (3) The outdated experiences may make the MARL system unstable. Therefore, the motivation of this work is to propose a cooperative MARL approach to address these problems. Specifically, the decentralized training with limited communication is applied to reduce communication latency in the proposed approach. It allows the neighbouring agents to share their local observations and local rewards to address partial observability issues. In the meantime, the local rewards between neighbouring agents are public. To get rid of the influence of coefficient changes on the reward function, this work uses the Shapley value [30] to generate a synthetic reward, i.e., the Shapley value reward function. The synthetic reward represents the contribution of agents in a team (an agent and its neighbours) for reducing traffic congestion. It can encourage cooperation between neighbours without any coefficient. Besides, to handle the outdated experiences, the Kullback-Leibler (KL) divergence of the current policy and the previous policy is applied to measure whether the experience is outdated. On this basis, the outdated experience is discarded to update the loss value. Finally, the proposed approach is evaluated under a synthetic traffic grid and three real-world traffic grids through comparisons with currently popular methods.

The main contributions of this work are: (1) The non-stationary of traffic signal environment is considered in this work, and it is handled by using KL divergence of the current policy and the previous policy. (2) A Shapley value reward and communication between neighbours are used to improve cooperation among agents. The Shapley value reward avoids the selection of reward coefficients. Communications between neighbours reduce information transmission time, compared with global information sharing. (3) A detailed performance analysis is provided. The experimental results show that, the proposed approach can reduce the average travel time of vehicles and make agent obtain higher accumulated rewards.

The rest of this work is arranged as follows. Section 2 provides related works in the field of traffic light control. Section 3 describes the background on RL. The framework and some necessary components of the system are introduced in Section 4. Section 5 describes the experiments for evaluating and analysing the performance of the proposed approach. Finally, this work is concluded in Section 6.

2. Related works

Traditionally, traffic signal control approaches are heuristic, including the fixed-time [21], longest-queue-first [39], and self-organizing methods [6], etc. The fixed-time method controls the traffic signal according to a fixed cycle about the red, green, and yellow signals. Every signal has a pre-set length of time. The main idea of the longest-queue-first method is preferentially setting the green signal in the direction with the longest queue. According to the results in the approach of [45], compared to highly dynamic and complex traffic environments, these methods have better performance under low traffic dynamic traffic environments. Besides, the self-organizing method needs a professional and experienced operator to set the parameters of the control program, which is inconvenient. The advantage of the proposed approach is that it can learn traffic signal control policy by interacting with the traffic environment.

The RL is applied to ATSC to deal with dynamic and complex traffic scenarios. The RL controls the traffic signal without any additional assumptions about the transition and distribution of the traffic environments. Early works using tabular Q-learning control traffic signals on isolated intersections [1,2]. The curse of dimensionality is an unavoidable defect of the tabular Q-learning. In recent years, The DRL adopts artificial neural networks to fit a complex RL model [22], thereby eliminating this defect. In [11], a deep-stacked autoencoder neural network for estimating the Q-function is introduced. The performances of deep policy-gradient methods and value-function-based methods under complex traffic environments are verified in [23]. The approach of [36] discusses the control

logic of the DRL model, and uses synthetic data and real-world data to verify the performances of the DRL model on different and dynamic traffic scenarios. These studies are all conducted on isolated intersections, with the limitations of environment assumptions and partial observability, thus their scalability is poor. In this paper, the proposed approach is extended to multiple intersections, and agents collaborate with each other.

In recent years, the application of DRL has been greatly developed in large-scale traffic environments. i.e., MARL. In a linear road scenario assuming that the state of the traffic environment is completely observable, the traffic signals are optimized by the Deep Q-Network (DQN) [19]. But this approach makes the state space grows exponentially with the number of intersections. Decentralized training, limited communication, and hierarchical structure are common methods to overcome this shortcoming. An actor-critic model combines with limited communication, neighbourhood fingerprints and a distance factor shows a good performance for controlling large-scale traffic signals [5]. Based on [5], the work in [20] further studied a hierarchical structure composed of managers and workers. In the approaches of [5] and [20], the importance of each part of the communication message is equal, i.e., the agent pays the same attention to each part of the communication message. In order to understand the importance of different parts of the communication message and to realize cooperation between neighbouring agents, a novel structure that combines graph attention network and communication is proposed in [35]. Besides, the reward function of these previous works is heuristic, leading to the high sensitivity of the performance of the DRL model [34]. A max-pressure reward function is introduced in [34], where the results show the max-pressure reward function is better than the heuristic reward function on arterial network scenarios. In the approach of [25], two attention models are used to achieve end-to-end training, and they can handle scenarios with different lanes and phases. The Nash Equilibrium is used to improve RL for ATSC, which are Nash Advantage Actor–Critic and Nash Asynchronous Advantage Actor–Critic [37]. A new traffic indicator, mixed pressure, in the approach of [8], is developed to analyze the impacts of stationary and moving vehicles on intersections. The intensity in [44] leads reward design and state representation to reflect the status of vehicles. Based on the max pressure [32], a concept named efficient pressure is proposed to present traffic movement in [38], and then the efficient max pressure method is designed to control traffic signals. Results in [38] show that the traffic state is an important factor for the ATSC methods with less training and lower complexity. Furthermore, the approach of [43] combines efficient pressure and effective running vehicles to build a method named advance max pressure, which is applied to control traffic signals effectively while taking both running and queuing vehicles into consideration. In the approach of [40], a hierarchical policy framework is used to control traffic signals, in which the local policy is learned to control signals and the high-level policy learns to cooperate with other agents.

The differences between this work and previous works are that this work enhances mutual cooperations between agents by using a Shapley value reward function to calculate the reward, and handles the non-stationary problem in the traffic signal control environment via the KL divergence.

3. Background

In this section, the background of this work is presented. Firstly, the deep Q-network algorithm is introduced in Section 3.1, which is the basis of this work. Secondly, in Section 3.2, the ATSC is formulated as a multi-agent system and the deep Q-network algorithm is extended to the multi-agent system (i.e. multi-agent reinforcement learning). Finally, the Shapley value is introduced in Section 3.3.

3.1. Deep Q-network

RL is one of the paradigms of machine learning. In RL, the agent interacts with the environment by trial and error at every time step t. Its goal is learning an optimal policy π to maximize the accumulated reward $G = \Sigma^{T}_{t=0}\gamma^{t}r_t$, where T is the end time of an episode, γ is a discount factor, and r_t is a scalar reward at time step t. The policy π is a distribution to specify the probability of taking the action a in each state s_t. A value function $Q^{\pi}(s_t, a_t)$ estimates the expected accumulated reward on a state-action pair (s_t, a_t) when following the policy π. It is described by

$$Q^{\pi}(s_t, a_t) = \mathbb{E}(G|s_t, a_t), \tag{1}$$

where $\mathbb{E}$ denotes the expected operator. The DQN approximates the value function $Q^{\pi}(s_t, a_t)$ by an evaluated network and a target network. The DQN is based on Q-learning, which estimates the value function as

$$Q^{\pi}(s_t, a_t) = Q^{\pi}(s_t, a_t) + \alpha(Y^{tg} - Q^{\pi}(s_t, a_t)), \tag{2}$$

where $Y^{tg} = r_t + \gamma \max_a Q^{\pi}(s_t, a)$ is the target value, $\max_a Q^{\pi}(s_t, a)$ represents selecting the maximal value under the state s_{t+1}. The DQN update the parameters of the evaluated network by the following loss function

$$L(\Theta) = \mathbb{E}[(r_t + \gamma \max_a Q^{t}(s_{t+1}, a|\Theta^{-}) - Q^{e}(s_t, a_t|\Theta))^2], \tag{3}$$

where $Q^{t}(s_{t+1}, a|\Theta^{-})$ is the estimated value by the target network, $Q^{e}(s_t, a_t|\Theta)$ is the estimated value by the evaluated network, Θ^{-} is the parameters of the target network, and Θ is the parameters of the evaluated network. The parameters Θ^{-} are updated by copying the parameters Θ after every C trained cycles, where C is a constant.

3.2. Multi-agent reinforcement learning

DQN is an individual agent model, which assumes the environment is stationary. This assumption is untenable in large-scale traffic scenarios [26]. MARL is an effective approach to solve complex tasks by the cooperation of individual agents.

Consider a multi-agent system, the agent $i \in \mathsf{F}$ only obtains a local observation o^i at time step t, where F is a set of agents. $s_t = \cup_{i=1,2,\ldots,m} o^i_t$ is the state of the system, where i denotes each agent, $m = |\mathsf{F}|$ is the number of agents. By using limited communication, an observation $O^i_t = o^i_t \cup o^{nei}_t$, where o^{nei}_t represents the local observation of the neighbour nei of the agent i. The agent i executes a local action a^i_t following policy π^i_t on the observation O^i_t. The environment feeds a local reward r^i_t to the agent i. Finding a joint action $U = \cup_{i\in\mathsf{F}} a^i_t$ to maximize the expected return is the goal of the MARL, and the system according to a transition function $P(s_{t+1}|s_t, U)$, where s_{t+1} is the new state of the system, and P is a state transition distribution. Similar to the observation, a reward $R^i_t = F(r^i_t, r^{nei}_t)$ by communicating with neighbours of the agent i, where F is a map, and r^{nei}_t is the local reward of the neighbour of the agent i. The value function of the agent i following policy π is represented by

$$Q^\pi_i(O^i_t, a^i_t) = Q^\pi_i(O^i_t, a^i_t) + \alpha(Y^{tg}_t - Q^\pi_i(O^i_t, a^i_t)), \tag{4}$$

where $Y^{tg}_t = R^i_t + \gamma \max_{ai} Q^\pi_i(O^i_{t+1}, a^i)$ is the target value, $\max_{ai} Q^\pi_i(O^i_{t+1}, a^i)$ represents selecting the maximal value under the observation O^i_{t+1}. When the value function is estimated by a deep artificial neural network, the agent i updates parameters of the evaluated network by the following loss function

$$L(\Theta^i) = \mathbb{E}[(Y^{tg}_t - Q^e(O^i_t, a^i_t|\Theta^i))^2], \tag{5}$$

where Y^{tg}_t is rewritten as $= R^i_t + \gamma \max_{ai} Q^t(O^i_{t+1}, a^i|\Theta^i_-)$ is the target value that is estimated by a deep artificial neural network, $Q^t(O^i_{t+1}, a^i|\Theta^i_-)$ is the estimated value by the target network of the agent i, $Q^e(O^i_t, a^i_t|\Theta^i)$ is the estimated value by the evaluated network of the agent i, Θ^i_- is the parameters of the target network, and Θ^i is the parameters of the evaluated network. The updated way of the parameters Θ^i_- is similar to the DQN.

3.3. Shapley value

Shapley value distributes the cooperation benefits fairly by considering the contributions made by each agent [29]. The Shapley value of agent i is the average value of i's expected contribution to a cooperative project. Given a cooperative game $\Gamma = (\square, v)$, where $\square$ is a set of the agents, and v is the secular equation of the contribution of each agent in this cooperative game. For any $C \subseteq \square \setminus i$, $\delta_i(\mathsf{C}) = v(\mathsf{C} \cup i) - v(\mathsf{C})$ is a marginal contribution, then the Shapley value of each agent i is described by

$$Sh_i(\Gamma) = \sum_{\mathsf{C}\subseteq\square\setminus\{i\}} \frac{|\mathsf{C}|!(|\square| - |\mathsf{C}| - 1)! \cdot \delta_i(\mathsf{C})}{|\square|!}. \tag{6}$$

4. Cooperative deep reinforcement learning for traffic signal control

This section describes the implementation details of the cooperative DRL approach in this work. Specifically, the traffic environment is formulated as an RL regime in Section 4.1. The deep neural network is described, and its hyper-parameters are presented in Section 4.2. Then, the loss function optimized by using the KL divergence, and Shapley value reward are introduced in Section 4.3 and 4.4, respectively. Note that the proposed approach is based on DQN, which is further beyond the simplistic variant of DQN. In the proposed approach, the DQN is extended to MAML, and then it combines with the optimized loss function and Shapley value reward for cooperations between agents.

4.1. Problem definition

This work regards the traffic environment as a traffic grid composed of multiple intersections. The movement of traffic flow is controlled by traffic signals at each intersection. The goal of ATSC is to reduce traffic congestion. Each intersection has four directions, which are North (N), South (S), East (E), and West (W). The edge of each direction consists of three lanes, which are a straight lane (sl), a left-turn lane (lt), and a right-turn lane (rt). A set L consists of sl, lt, and rt, i.e., $L = (sl, lt, rt)$. A traffic signal controls a lane, thus a arrangement of traffic signals constitutes a phase. The phase allows nonconflicting movements. Each traffic signal has three statuses: red, green, and yellow.

In a given traffic grid, each intersection has an agent to control traffic signals. Therefore, each agent has at least 2 neighbours and at most 4 neighbours. The neighbours of agent i are denoted as a set $\square_i = (N_i, S_i, W_i, E_i)$. Using e^{dir}_{ij} represents a directed lane, where i denotes the depart intersection, j denotes the objective intersection, dir is one of sl, lt, and rt. To prevent the phase from changing too often, the phase changes after Δt time steps. The local observation o^i_t of the agent i consists of the queue length q^{dir}_{ji} and the number of vehicles n^{dir}_{ji} on each incoming lane at time step t. The queue length q^{dir}_{ji} represents the number of vehicles waiting on the lane e^{dir}_{ji}. The observation of agent i finally denotes as $O^i_t = \{o^k_t\}_{k\in\square_i \cup i}$. The global state $s_t = \{o^k_t\}_{k\in\mathsf{F}}$, where F is the set of agents.

The local reward r^i_t of the agent i is the negative average queue length of all incoming lanes, i.e., $r^i_t = -\left(\sum_{dir\in L}\sum_{j\in\square_i} q^{dir}_{ji}\right)/(|\square_i| \cdot |L|)$. To achieve cooperation, the reward $R^i_t = F(r^i_t, \{r^k_t\}_{k\in\square_i})$, where F is a specific map. Naturally, the local action of the agent i is defined

as a possible phase, which is the order combination of traffic signals. The action space of the agent i contains all possible phases.

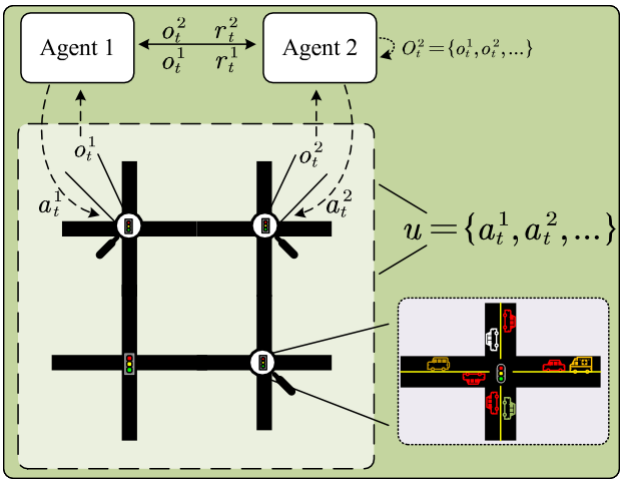

Fig. 1. MARL of cooperative traffic signal control system.

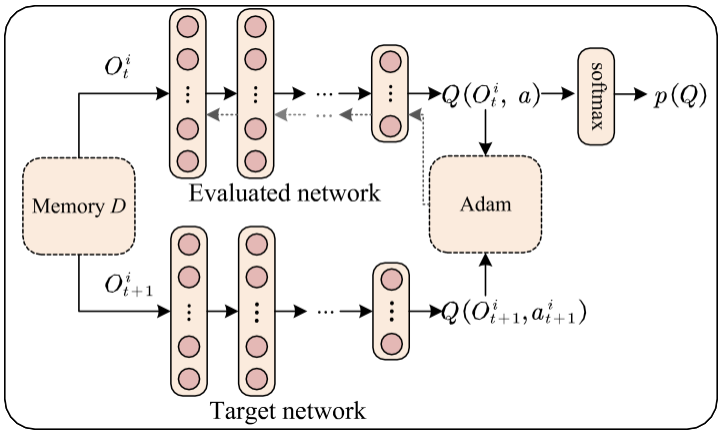

Fig. 2. Deep Neural Network.

Finally, after formulating the various elements of the transportation system, ATSC can be formulated in a standard RL regime. The transition of ATSC is shown in Fig. 1. At time step t, the agent i obtains a local observation o_t^i. By communicating with neighbours, the observation O_t^i is obtained, and then the agent i executes an action a_t^i on observation O_t^i. The transportation system transfers to a new state $s_{t+1} \sim P(s_t, U)$, where P is a state transition distribution, $\mathbf{T} = a_t^i$ is a joint action of all agents, where $i \in \mathbf{F}$. Meanwhile, a local reward r_t^i is obtained by i.

4.2. Deep neural network settings

Similar to DQN, the value function of each agent is estimated by an evaluated network and a target network. The observation O_t^i is a vector of the queue length and the number of vehicles. The vector length is related to the number of incoming lanes and the number of neighbours. Therefore, the Deep Neural Network (DNN) is a multi-layer perceptron. The nonlinear processing capability of the system is achieved by using rectifier linear units between each layer. As Fig. 2 shows, the value $Q^{\pi}(O_t^i, a_t^i|\Theta^i)$ is estimated by the evaluated network with parameters Θ, and the value of next observation-action pair $Q^{\pi}(O_{t+1}^i, a_{t+1}^i|\Theta^i)$ is estimated by the target network with parameters Θ^-. The evaluated network and the target network are fed the observation and the next observation, respectively. Both networks output a vector, which has dimension as same as the size of the action space, i.e., the output is not about a specific state-action pair value, but about all state-action pair values. The evaluated network is trained by using stochastic gradient descent, which is introduced in Section 4.3. Due to that the target network aims to ensure the target value of the evaluated network changes as little as possible, the updated method of the target network is soft-update [12] which is different to the original DQN. It can be denoted as

$$\Theta^{i-} = (1-\tau)\Theta^{i-} + \tau\Theta^i, \tag{7}$$

where τ is a constant coefficient, and $\tau \ll 1$. Besides, the last layer of the evaluated network is a soft-max layer, which maps the $Q_i^{\pi}(O_t^i, a_t^i|\Theta^i)$ to a distribution $p(Q_i^{\pi})$. In this work, the evaluated network has two hidden layers, which are 100 and 50 units, respectively. The output layer has four units, which is the same as the action space. The target network is the same as the evaluated network.

4.3. The optimized loss function

In this work, the experience replay technology is applied to train DNN. A transition experience set $e_t = (O_t^i, a_t^i, R_t^i, O_{t+1}^i, p(Q_i^{\pi}))$ is collected by agent i at every time step, and the set is saved into an experience memory D. A mini-batch of experiences is sampled randomly from the memory D when the DNN is trained. The time steps of generating experiences are not always close to the training time steps. In other words, the time step interval between generating and using experiences may exceed one training period, or even more than ten training periods. The maximal time interval is determined by the length of the memory D and training period. A larger time interval means more DNN training times, and possibly a larger difference between the experience policy and the current policy. As discussed in Section 1, a huge difference in policies can mislead the agent's learning, because outdated experience is also adopted by the agent to adjust its policy.

In this work, the difference is measured by the KL divergence between the old policy and the current policy, it is described as

$$d_i = KL(p(Q_i^{\pi_{old}}), p(Q_i^{\pi})) = \sum_{l=1}^{N} [p^l(Q_i^{\pi_{old}}) \log(\frac{p^l(Q_i^{\pi_{old}})}{p^l(Q_i^{\pi})})], \tag{8}$$

where N is the action space, and π_{old} represents the policy of the experience (the $p(Q_i^{\pi})$ in the experience is denoted as $p(Q_i^{\pi_{old}})$ when it is sampled the memory D). Then the different d_i is used to adjust Eq. (5), thus the loss function is described as

$$L(\Theta^i) = \mathbb{E}[(\beta e^{-d_i}(Y_t^{tg} - Q^e(O_t^i, a_t^i|\Theta^i))^2], \tag{9}$$

where $Y_t^{tg} = R_t^i + \gamma \max_a Q^t(O_{t+1}^i, a^i|\Theta_-^i))$ is the state update target value, e is Euler-number, e^{-d_i} is an inverse function and less than one, it does not cause the divergence of the equation.

4.4. Shapley value reward

In order to reflect the effectiveness of the policy for reducing traffic congestion, agents not only consider their own local rewards, but also the local rewards of neighbours. A simple way is to construct a linear equation for the local rewards of oneself and neighbours. As discussed in [34], it is very sensitive to coefficients if the reward function is a linear equation, which will have a great influence on the performance of DRL. Thus, the reward is generated by the Shapley value in this work. The equation v is defined as the expected reward of all neighbouring agents, i.e., $v(C) = \mathbb{E}(\{r^k\}_{k\in C})$, thus $\delta_i(C) = \mathbb{E}(\{r^k\}_{k\in C\cup\{i\}}) - \mathbb{E}(\{r^k\}_{k\in C})$. Then the reward signal is calculated by Eq. (6). This reward function represents the contribution of agent i for reducing traffic congestion and the goal of agent is to maximize its cumulative reward (i.e. the agent i maximizes its contribution to the team). The policy of the agent is dynamically adjusted by considering the neighbours' reward, which can encourage the agent to achieve cooperation with its neighbours.

5. Results

In order to test the performance of the proposed approach, experiments are performed in a simulation platform CityFlow using public datasets. The results include ablation study analysis and performance comparisons with existing methods. The simulation platform and the structure of intersections is presented in Section 5.1. The data statistics of datasets and the parameters setting of agents are described in Section 5.2. The ablation studies in the static and dynamic traffic dataset are presented, and performances are analysed in Section 5.3. In Section 5.4, the results of comparison with existing methods are provided.

5.1. Simulation platform

The proposed approach is tested and verified in different scale traffic grids. To test the effect of the loss function and Shapley value reward, a synthetic traffic grid including 2×2 intersections is constructed by simulation of urban mobility [10]. The simulation of urban mobility is an open-source simulation platform, and it provides a series of python application programming interfaces to get traffic information and control traffic signals. Three large-scale traffic grids are constructed by CityFlow [42], which is an open-source traffic simulation platform and supports large-scale city traffic signal control.

5.2. Datasets and parameters setting

In the experiments, two synthetic traffic datasets are applied to a synthetic traffic grid, and their configuration details are shown in Table 1 and 2. In the static dataset, vehicles arrive at each intersection through a random process. In the simulation process, the average

arrival rate is 0.2, i.e., two vehicles arrive at the entrance every ten seconds. The turning ratios at the intersection are set to 60% (straight), 20% (left), and 20% (right). In the dynamic dataset, an episode is divided into six parts, and each part has 600

Table 1
The static traffic dataset.

Arrival rate(vehicles/s)	Straight rate	Left rate	Right rate	Start time (s)	End time (s)
0.2	0.6	0.2	0.2	0	3600

Table 2
The dynamic traffic dataset.

Arrival rate(vehicles/s)	Straight rate	Left rate	Right rate	Start time (s)	End time (s)
0.100	0.600	0.200	0.200	0	600
0.168	0.600	0.200	0.200	601	1200
0.200	0.600	0.200	0.200	1201	1800
0.140	0.600	0.200	0.200	1801	2400
0.120	0.600	0.200	0.200	2401	3000
0.100	0.600	0.200	0.200	3001	3600

Table 3
Data statistics of the real-world dataset.

Dataset	intersections	Arrival rate (vehicles/300 s)			
		Mean	std	Max	Min
$D_{NewY\ ork}$	196	240.79	10.08	274	216
$D_{Hangzhou}$	16	526.63	86.70	676	256
D_{Jinan}	12	250.70	38.21	335	208

Table 4
The parameters of the proposed method.

Parameter	Value	Parameter	Value
Batch size	64	ϵ for exploration	0.01
Learning rate	1e-3	Target network update τ	0.05
Memory length	5000	begin training	3600
Δt	5 s		

seconds. In addition, three large-scale real-world traffic grids and three traffic dataset are used to test the proposed method [35]. These three large-scale real-world traffic grids include 12 intersections, 16 intersections, and 196 intersections, respectively. The 12 intersections traffic grid extracted from Jinan, the 16 intersections traffic grid extracted from Hangzhou, and the 196 intersections traffic grid extracted from New York. Correspondingly, the datasets are real-world traffic data from three cities: Jinan, Hangzhou, and New York. The data statistics (including mean and the standard deviation (std)) of the real-world datasets are listed in Table 3, and the detailed descriptions of the datasets can be obtained in [35]. The parameters of each agent are shown in Table 4.

5.3. Ablation study

In this experiment, the proposed method is evaluated by using the static and dynamic traffic dataset. The static traffic dataset is used to test the effects of the Shapley value reward and the optimized loss function in an ablation study. The algorithms are described in detail as follows:

1) Independent DQN (IDQN): It applies DQN to each intersection without any communication and other optimization.
2) Limited Communication DQN (LCDQN): This approach is that the agent communicates with its neighbours based on IDQN.
3) Shapley Value reward DQN (SVDQN): The Eq. (6) is used to calculate the reward signal in this approach, and other settings are the same as LCDQN.
4) Optimized Loss function DQN (OLDQN): The loss function of this approach is Eq. (9), and others are the same as Shapley value reward DQN.

The synthetic traffic grid environment and static dataset are applied in the experiment, and the average travel time is used as the evaluation metric. The training results are listed in Table 5. The standard deviation and the average travel time are calculated by the last 20 training results. From the results, the LCDQN reduces the average travel time than IDQN by using limited communication, and the standard deviation is also reduced by 40.9%. It indicates that limited communication can effectively improve the stability of the system, i.e., it can solve the problem of partially observable. The SVDQN further improves the performance of the system by using Shapley value reward, that is, reduces the average travel time of the system. This result also shows that the Shapley value reward can promote cooperation between agents. The OLDQN has a better performance compared with SVDQN, 1.4% reduction of

Table 5
The average travel time of different variants in the synthetic traffic dataset.

	IDQN	LCDQN	SVDQN	OLDQN
Average travel time (s)	167.59	156.95	145.89	143.91
std	2.74	1.62	1.62	0.78

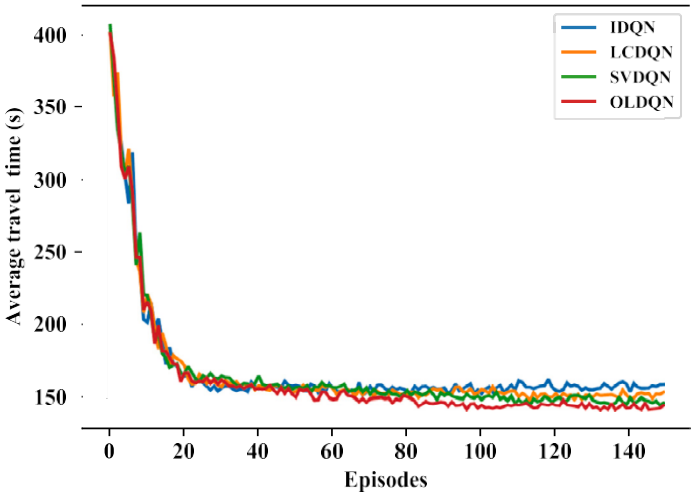

Fig. 3. Comparison between different variants of the proposed approach.

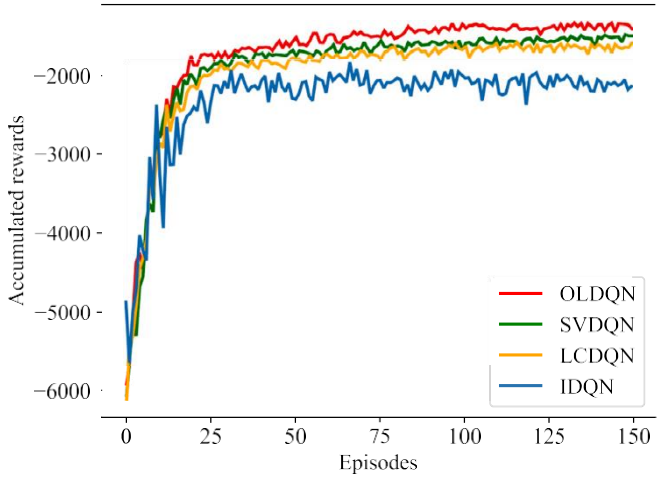

Fig. 4. The accumulated rewards of different variants of the proposed approach.

the average travel time and 51.9% reduction of the standard deviation. The optimized loss function restricts outdated experiences to make the system stable. The OLDQN reduces the average travel time by 14.1% and the standard deviation by 71.5% compared with the original IDQN. These results show that the Shapley value reward and the optimized loss function play key roles in improving the performance of the proposed approach. Fig. 3 illustrates the performance of these variants during training. The curve of OLDQN is at the bottom, which indicates that OLDQN is better than other variants.

The goal of RL is to maximize the accumulated reward of the agent. The change in cumulative rewards can reflect whether the quality of the policy has improved. Fig. 4 shows the accumulated rewards of agents in the synthetic traffic grid environment and static dataset. The curves are the sum of four agents, which illustrates the collaboration of four agents. Compared with IDQN, the LCDQN has a significant increase in the accumulated reward. This indicates that communicating with neighbours is beneficial for enhancing collaboration between agents. The SVDQN obtains a higher cumulative reward and earlier than LCDQN. The difference between SVDQN and LCDQN is that the SVDQN adopts the Shapley value reward. Thus, the result shows that the Shapley value reward can facilitate collaboration among agents. The OLDQN receives the highest accumulated reward, which the OLDQN eliminates the effect of outdated experiences by using KL divergence. The fluctuations in IDQN indicate that its policy changes greatly, and it is difficult

Table 6
The average travel time of different variants in the dynamic traffic dataset.

	SVDQN	OLDQN	GRDQN
Average travel time (s)	179.02	177.22	193.27
std	4.91	3.70	8.01

Table 7
The average travel time of all methods.

Model	New York	Hangzhou	Jinan
Fixedtime [7]	1950.27	728.79	869.85
MaxPressure [32]	1633.41	422.15	361.33
CGRL [27]	2187.12	1582.26	1210.70
Individual RL [36]	–	345	325.56
GCN [24]	1876.37	768.43	625.66
CoLight-node [35]	1493.37	331.50	340.70
CoLight [35]	1459.28	297.26	291.14
This work	1054.10	306.39	285.59

– represents no results provided.

to find a better policy in a short period. However, the SVDQN can find a policy to obtain the highest accumulated reward and keep it smooth.

Table 6 lists the training results of dynamic traffic dataset. The GRDQN in Table 6 represents that the reward function of DQN is a global reward, which is an average value of the local reward of all agents in the traffic grid. Results show that the OLDQN has a lower standard deviation than other methods, i.e. it is more stable than SVDQN and GRDQN in dynamic traffic environment. The standard deviation of OLDQN is lower than the SVDQN in synthetic and dynamic dataset, as the optimized loss function can reduce volatility. The rewards of the SVDQN and GRDQN are the Shapley value and global reward, respectively. The SVDQN reduces the average travel time by 7.37% compared to the GRDQN. This shows that the Shapley value reward is more effective than global reward in reducing traffic congestion. In addition, for the GRDQN every agent needs to communicate with others while obtaining the global reward, which leads to more communication costs compared to the SVDQN.

5.4. Comparison with existing methods

In this experiment, the proposed approach (i.e., OLDQN) is tested and verified in three real-world traffic datasets. The baseline approaches as follows:

1) Fixedtime [7]: Fixed-time is a pre-set plan for cycle and phase time. In multiple intersections environment, it with an offset.
2) MaxPressure [32]: This method is a network-level traffic signal control method, which chooses the phase with the maximum pressure greedily.
3) CGRL [27]: A RL model for finding an optimal joint action of multiple agents. This model uses a coordination graph to achieve cooperation between agents.
4) Individual RL [36]: This method uses the DRL model with a novel neural network structure, and it has an experience palace structure to address the data imbalance problem. Besides, there is no sharing of traffic information between agents.
5) GCN [24]: A DRL model controls traffic signals and uses a graph convolutional neural network to extract the feature of the neighbours.
6) CoLight [35]: A DRL model uses a graph attentional network to learn communication, and it uses geo-distance to determine its neighbours.
7) CoLight-node [35]: It is the same as CoLight, except the neighbours are determined by a node distance.

Table 7 lists the average travel time of the proposed approach and all baseline approaches. The Fixedtime is clumsy, it cannot adaptively control the traffic signals. Thus, the proposed approach has a great improvement compared with the Fixedtime. The best improvement is a 67.2% reduction of the average travel time in Jinan. Even in New York, the proposed approach has a 46.0% reduction of the average travel time. The MaxPressure method is a transportation method, and it only considers the current traffic situation, without learning previous experiences and predicting the future situations to improve its control rule. The gap between it and the proposed approach becomes larger as the network expands. The gap is 21.0% in Jinan, but it is 35.5% in New York. The reasons are that the proposed approach has learned previous experiences and cooperation with the neighbours. The CGRL learns a joint action to achieve cooperation, which makes the action space larger, resulting in a very large traveling time. The Individual RL is selfish and it is difficult to achieve the desired result. The Colight-node uses node distance to determine the neighbours, which ignores the effect of different geo-distance. Thus, it is worse than the proposed approach in all traffic grids. In small-scale traffic grids, the performance of CoLight is comparable to the proposed approach. However, in a large-scale traffic grid, the collaboration ability of CoLight is significantly worse than the proposed approach. Because the proposed approach uses the Shapley value reward and the optimized loss function to encourage collaboration among agents.

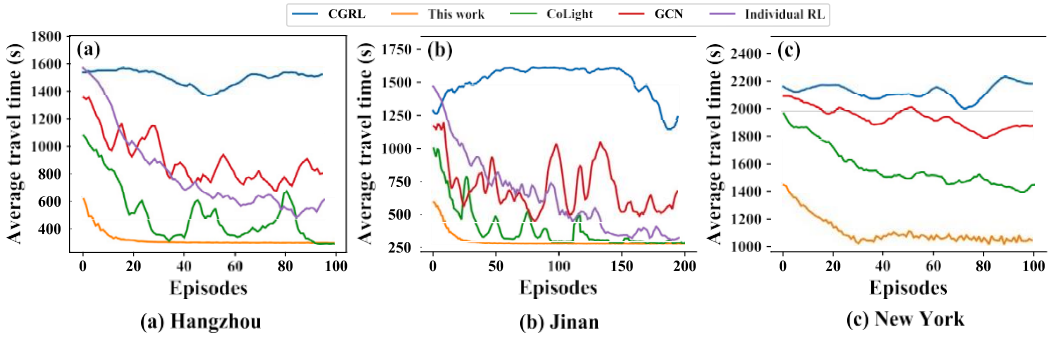

Fig. 5. Convergence speed of the proposed approach and other four RL baselines during training. (a) is training in Hangzhou dataset, (b) is training in Jinan dataset, and (c) is training in New York's dataset.

Fig. 5 illustrates the convergence speed of the proposed approach and other RL baselines. Although the final result of the proposed approach is worse than the CoLight in the Hangzhou traffic scenario, the convergence speed of the proposed approach is faster than the CoLight, and it is more stable than the CoLight. The proposed approach has a lower average travel time in each traffic grid and is more stable compared with other baselines. The CGRL convergence is very poor in each traffic grid. The GCN is fluctuating in Hangzhou and Jinan.

5.5. Comparisons with non-DQN methods

In this subsection, the comparisons with the recent non-DQN algorithms including actor–critic based algorithm and hierarchical RL based algorithm [40,46] are presented. In the approach of [40], a hierarchical, cooperative, and multi-critic RL method is proposed, namely HiLight, which is based on LocalCritic and NBHDCritic. The differences between LocalCritic, NBHDCritic, and HiLight are that the LocalCritic uses only one critic for local traffic travel time, the NBHDCritic uses only one critic for neighbourhood travel time, and HiLight has two critics for local and neighbourhood travel times. These three methods (HiLight, LocalCritic and NBHDCritic) are verified in the Hangzhou and Jinan traffic environments [40], which are the same as the traffic datasets used in Section 5.4. The average travel time of LocalCritic, NBHDCritic and HiLight are 343, 516, and 256 seconds in Hangzhou, respectively. Compared with LocalCritic and NBHDCritic, the proposed approach has a lower travel time (306.39 seconds) due to that the LocalCritic and NBHDCritic only consider local control policy or cooperative policy leading to higher average travel times. Compared with the HiLight, the proposed approach has a higher average travel time. However, the proposed approach has a lower average travel time in Jinan, which is 285.59 seconds and the HiLight has an average travel time of 290 seconds. In addition, the HiLight has a more complex structure than the proposed approach, while the former has multi sub-policies (in local traffic signal control) and a multi- critic controller, and the latter only has an evaluated and target network. In addition, auto-learning communication reinforcement learning (ALCORL) in [46] is also used for comparisons, which is based on the advantage actor–critic algorithm. The ALCORL uses an autoencoder to learn communication messages, which enhances cooperation by receiving messages from neighbouring intersections. The ALCORL is tested in the Hangzhou dataset, and it has an average travel time of 315.87 seconds. The proposed approach is 306.39 seconds (an improvement of 3.1% is achieved). In a summary, these comparison results with actor-critic-based methods [40,46] show that the proposed approach has some advantages in the Hanghzou and Jinan traffic scenarios.

6. Conclusion

In this work, an ATSC approach based on MARL is proposed to reduce the travel times of vehicles. It is based on the framework of limited communication decentralized training, uses the KL divergence between current and previous policy to update the loss value, and uses Shapley value reward to encourage the cooperation of neighbouring agents. Each component of the proposed approach is first verified on the synthetic dataset. The result shows that the optimized loss function can effectively reduce the standard deviation of the travel time, and the Shapley value reward can significantly reduce the average travel time. In addition, the proposed method is evaluated by using real-world traffic datasets including Hangzhou, Jinan, and New York. Results demonstrate that this work can control traffic signals well even in a large-scale traffic grid with 196 intersections, e.g. compared with other DRL methods the average travel time of the New York dataset experiment is reduced by 27.8%. However, the limitation of this work is that the computational complexity of the Shapley value reward is high. It is worth noting that this only affects the proposed approach in the training stage. In future work, how to optimize the computational complexity of Shapley value reward will be further studied. Furthermore, the proposed approach will be further verified in the city-level traffic environment.

CRediT authorship contribution statement

Junxiu Liu: Conceptualization, Formal analysis, Validation, Writing – original draft. **Sheng Qin:** Conceptualization, Methodology, Software, Writing – original draft, Writing – review & editing. **Min Su:** Validation, Writing – review & editing. **Yuling Luo:** Validation,

Visualization, Writing – review & editing. **Yanhu Wang:** Validation, Writing – review & editing. **Su Yang:** Validation, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

Acknowledgements

This research is supported by the National Natural Science Foundation of China under Grant 61976063, the Guangxi Natural Science Foundation under Grant 2022GXNSFFA035028, research fund of Guangxi Normal University under Grant 2021JC006, the AI+Education research project of Guangxi Humanities Society Science Development Research Center under Grant ZXZJ202205.

References

[1] B. Abdulhai, R. Pringle, G.J. Karakoulas, Reinforcement learning for true adaptive traffic signal control, J. Transp. Eng. 129 (2003) 278–285.
[2] P.G. Balaji, X. German, D. Srinivasan, Urban traffic signal control using reinforcement learning agents, IET Intell. Transp. Syst. 4 (2010) 177–188.
[3] H. Ceylan, M.G. Bell, Traffic signal timing optimisation based on genetic algorithm approach, including drivers' routing, Transp. Res., Part B, Methodol. 38 (2004) 329–342.
[4] C. Chen, H. Wei, N. Xu, G. Zheng, M. Yang, Y. Xiong, K. Xu, Z. Li, Toward a thousand lights: decentralized deep reinforcement learning for large-scale traffic signal control, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2020, pp. 3414–3421.
[5] T. Chu, J. Wang, L. Codeca, Z. Li, Multi-agent deep reinforcement learning for large-scale traffic signal control, IEEE Trans. Intell. Transp. Syst. 21 (2020) 1086–1095.
[6] S.B. Cools, C. Gershenson, B. D'Hooghe, Self-organizing traffic lights: a realistic simulation, in: Advanced Information and Knowledge Processing, Springer, 2013, pp. 45–55.
[7] K.G. F, Traffic Signal Timing Manual, Federal Highway Administration, United States, 2008.
[8] Z. Fang, F. Zhang, T. Wang, X. Lian, M. Chen, Monitorlight: reinforcement learning-based traffic signal control using mixed pressure monitoring, in: Proceedings of the 31st ACM International Conference on Information & Knowledge Management, 2022, pp. 478–487.
[9] J. García-Nieto, E. Alba, A. Carolina Olivera, Swarm intelligence for traffic light scheduling: application to real urban areas, Eng. Appl. Artif. Intell. 25 (2012) 274–283.
[10] D. Krajzewicz, J. Erdmann, M. Behrisch, L. Bieker, Recent development and applications of sumo - simulation of urban mobility, Int. J. Adv. Syst. Meas. 5 (2012) 128–138.
[11] L. Li, Y. Lv, F.Y. Wang, Traffic signal timing via deep reinforcement learning, IEEE/CAA J. Autom. Sin. 3 (2016) 247–254.
[12] T.P. Lillicrap, J.J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra, Continuous control with deep reinforcement learning, arXiv preprint, arXiv:1509.02971, 2015.
[13] J. Liu, M. Li, Y. Luo, S. Yang, S. Qiu, Human body posture recognition using wearable devices, in: 28th International Conference on Artificial Neural Networks (ICANN), 2019, pp. 326–337.
[14] J. Liu, T. Sun, Y. Luo, S. Yang, Y. Cao, J. Zhai, An echo state network architecture based on quantum logic gate and its optimization, Neurocomputing 371 (2020) 100–107.
[15] J. Liu, T. Sun, Y. Luo, S. Yang, Y. Cao, J. Zhai, Echo state network optimization using binary grey wolf algorithm, Neurocomputing 385 (2020) 310–318.
[16] J. Liu, G. Wu, Y. Luo, S. Qiu, S. Yang, W. Li, Y. Bi, EEG-based emotion classification using a deep neural network and sparse autoencoder, Front. Syst. Neurosci. 14 (2020) 1–14.
[17] J. Liu, J. Zhang, Y. Luo, S. Yang, J. Wang, Q. Fu, Mass spectral substance detections using long short-term memory networks, IEEE Access 7 (2019) 10734–10744.
[18] R. Liu, J. Zou, The effects of memory replay in reinforcement learning, in: 56th Annual Allerton Conference on Communication, Control, and Computing, 2018, pp. 478–485.
[19] X.Y. Liu, Z. Ding, S. Borst, A. Elwalid, Deep reinforcement learning for intelligent transportation systems, arXiv preprint, arXiv:1812.00979, 2018.
[20] J. Ma, F. Wu, Feudal multi-agent deep reinforcement learning for traffic signal control, in: Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS), 2020, pp. 816–824.
[21] A.J. Miller, Settings for fixed-cycle traffic signals, J. Oper. Res. Soc. 14 (1963) 373–386.
[22] V. Mnih, K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, M.G. Bellemare, A. Graves, M. Riedmiller, A.K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, D. Hassabis, Human-level control through deep reinforcement learning, Nature 518 (2015) 529–533.
[23] S.S. Mousavi, M. Schukat, E. Howley, Traffic light control using deep policy-gradient and value-function-based reinforcement learning, IET Intell. Transp. Syst. 11 (2017) 417–423.
[24] T. Nishi, K. Otaki, K. Hayakawa, T. Yoshimura, Traffic signal control based on reinforcement learning with graph convolutional neural nets, in: IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC, 2018, pp. 877–883.
[25] A. Oroojlooy, M. Nazari, D. Hajinezhad, J. Silva, Attendlight: universal attention-based reinforcement learning model for traffic signal control, Adv. Neural Inf. Process. Syst. 33 (2020) 4079–4090.
[26] G. Papoudakis, F. Christianos, A. Rahman, S.V. Albrecht, Dealing with non-stationarity in multi-agent deep reinforcement learning, arXiv preprint, arXiv: 1906.04737, 2019.
[27] E. van der Pol, F.A. Oliehoek, Coordinated deep reinforcement learners for traffic light control, in: NIPS'16 Workshop on Learning, Inference and Control of Multi-Agent Systems, 2016.
[28] D. Renfrew, X.H. Yu, Traffic signal control with swarm intelligence, in: 5th International Conference on Natural Computation (ICNC), 2009, pp. 79–83.
[29] L.S. Shapley, A value for n-person games, in: Contributions to the Theory of Games, Princeton University Press, 1953, pp. 307–318.
[30] L.S. Shapley, Stochastic games, Proc. Natl. Acad. Sci. 39 (1953) 1095–1100.
[31] M. Tan, Multi-agent reinforcement learning: independent vs. cooperative agents, in: Machine Learning Proceedings, 1993, pp. 330–337.

[32] P. Varaiya, The max-pressure controller for arbitrary networks of signalized intersections, in: Advances in Dynamic Network Modeling in Complex Transportation Systems, Springer, 2013, pp. 27–66.
[33] X. Wang, L. Ke, Z. Qiao, X. Chai, Large-scale traffic signal control using a novel multiagent reinforcement learning, IEEE Trans. Cybern. 51 (2021) 174–187.
[34] H. Wei, C. Chen, G. Zheng, K. Wu, V. Gayah, K. Xu, Z. Li, Presslight: learning max pressure control to coordinate traffic signals in arterial network, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2019, pp. 1290–1298.
[35] H. Wei, N. Xu, H. Zhang, G. Zheng, X. Zang, C. Chen, W. Zhang, Y. Zhu, K. Xu, Z. Li, CoLight: learning network-level cooperation for traffic signal control, in: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 2019, pp. 1913–1922.
[36] H. Wei, G. Zheng, H. Yao, Z. Li, IntelliLight: a reinforcement learning approach for intelligent traffic light control, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 2496–2505.
[37] Q. Wu, J. Wu, J. Shen, B. Du, A. Telikani, M. Fahmideh, C. Liang, Distributed agent-based deep reinforcement learning for large scale traffic signal control, Knowl.-Based Syst. 241 (2022) 108304.
[38] Q. Wu, L. Zhang, J. Shen, L. Lü, B. Du, J. Wu, Efficient pressure: improving efficiency for signalized intersections, arXiv preprint, arXiv:2112.02336, 2021.
[39] R. Wunderlich, C. Liu, I. Elhanany, T. Urbanik, A novel signal-scheduling algorithm with quality-of-service provisioning for an isolated intersection, IEEE Trans. Intell. Transp. Syst. 9 (2008) 536–547.
[40] B. Xu, Y. Wang, Z. Wang, H. Jia, Z. Lu, Hierarchically and cooperatively learning traffic signal control, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2021, pp. 669–677.
[41] K.L.A. Yau, J. Qadir, H.L. Khoo, M.H. Ling, P. Komisarczuk, A survey on reinforcement learning models and algorithms for traffic signal control, ACM Comput. Surv. 50 (2017) 1–38.
[42] H. Zhang, Y. Ding, W. Zhang, S. Feng, Y. Zhu, Y. Yu, Z. Li, C. Liu, Z. Zhou, H. Jin, CityFlow: a multi-agent reinforcement learning environment for large scale city traffic scenario, in: Proceedings of the World Wide Web Conference, 2019, pp. 3620–3624.
[43] L. Zhang, Q. Wu, J. Shen, L. Lü, B. Du, J. Wu, Expression might be enough: representing pressure and demand for reinforcement learning based traffic signal control, in: International Conference on Machine Learning, PMLR, 2022, pp. 26645–26654.
[44] W. Zhao, Y. Ye, J. Ding, T. Wang, T. Wei, M. Chen, Ipdalight: intensity- and phase duration-aware traffic signal control based on reinforcement learning, J. Syst. Archit. 123 (2022) 102374.
[45] G. Zheng, X. Zang, N. Xu, H. Wei, Z. Yu, V. Gayah, K. Xu, Z. Li, Diagnosing reinforcement learning for traffic signal control, arXiv:1905.04716, http://arxiv.org/abs/1905.04716, 2019.
[46] R. Zhu, W. Ding, S. Wu, L. Li, P. Lv, M. Xu, Auto-learning communication reinforcement learning for multi-intersection traffic light control, Knowl.-Based Syst. (2023) 110696.