PGA: An Efficient Adaptive Traffic Signal Timing Optimization Scheme Using Actor-Critic Reinforcement Learning Algorithm

Si Shen[1,2], Guojiang Shen[1*], Yang Shen[1], Duanyang Liu[1], Xi Yang[1], Xiangjie Kong[1]
[1] College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China
[e-mail: {gjshen1975, ldy, xyang}@zjut.edu.cn, yangshen014@outlook.com, xjkong@ieee.org]
[2] Department of Forensic Science, Zhejiang Police College, Hangzhou 310053, China
[e-mail: shengsi@zjjcxy.cn]
*Corresponding author: Guojiang Shen

Received August 19, 2020; revised October 20, 2020; accepted November 6, 2020; published November 30, 2020

Abstract

Advanced traffic signal timing method plays very important role in reducing road congestion and air pollution. Reinforcement learning is considered as superior approach to build traffic light timing scheme by many recent studies. It fulfills real adaptive control by the means of taking real-time traffic information as state, and adjusting traffic light scheme as action. However, existing works behave inefficient in complex intersections and they are lack of feasibility because most of them adopt traffic light scheme whose phase sequence is flexible. To address these issues, a novel adaptive traffic signal timing scheme is proposed. It's based on actor-critic reinforcement learning algorithm, and advanced techniques proximal policy optimization and generalized advantage estimation are integrated. In particular, a new kind of reward function and a simplified form of state representation are carefully defined, and they facilitate to improve the learning efficiency and reduce the computational complexity, respectively. Meanwhile, a fixed phase sequence signal scheme is derived, and constraint on the variations of successive phase durations is introduced, which enhances its feasibility and robustness in field applications. The proposed scheme is verified through field-data-based experiments in both medium and high traffic density scenarios. Simulation results exhibit remarkable improvement in traffic performance as well as the learning efficiency comparing with the existing reinforcement learning-based methods such as 3DQN and DDQN.

Keywords: Traffic signal timing, reinforcement learning, actor-critic, proximal policy optimization, generalized advantage estimation

This work was partially supported by the National NSFC under Grant 62073295, 61603339 and 62072409, Zhejiang Provincial NSFC under Grant LY20F030018 and LR21F020003, and Zhejiang Public Welfare Technology Research Program under Grant LGG19F030012.

http://doi.org/10.3837/tiis.2020.11.002 **ISSN : 1976-7277**

1. Introduction

Traffic signal timing (TST) is one of the fundamental problems in traffic engineering. It is of great importance to design and implement advanced signal timing scheme to reduce traffic congestion and pollution, and to achieve optimization and coordination of urban traffic. In recent decades, the blooming development of artificial intelligence makes the remarkable progress of modern intelligent transportation systems (ITS), e.g., intelligent signal control [1], big data analysis[2], [3], traffic flow prediction [4], pedestrian detection [5], etc.

As a typical kind of machine learning schemes and algorithms, reinforcement learning (RL) enables an agent to achieve sequential decisions through interactions with environment so as to maximize the objective reward function in a trail-and-error manner [6]. It has become a promising research direction of TST owing to its excellent interactiveness and adaptability with the dynamically changing traffic environment.

Early researches of RL-based TST method [7]–[9] mainly adopt tabular-based RL algorithms, which suffer from dimension curse in complicated traffic environment. Deep reinforcement learning (DRL) methods efficiently solve this problem by using deep neural networks to approximate state (or state-action) value functions, and present superior adaptability and stability in simulation experiments [10]–[12].

However, there still exists certain issues in DRL-based TST problem deserving further consideration. First and foremost, the traditional modeling methods are lack of efficiency in TST problem. Most existing works are value function-based DRL methods which relies on certain value functions that are hard to be appropriately defined or approximated in complex traffic scenarios. Meanwhile, the computational complexity of the methods are highly related to the dimensions of the defined state space and action space. The commonly used state representation such as discrete traffic state encoding (DTSE) [13] may significantly increase the dimensionality so as the computational complexity.

From field-application perspective, existing works are lack of concerns about security and feasibility of the proposed methods. Particularly, the phase sequence is assumed to be adjustable in most of the proposed traffic timing schemes, which means all available traffic phases for the studied intersection are activated in a random sequence according to the agent's action. Drivers and pedestrians around the intersection can't predict the next activated traffic phase. Therefore, the traffic signal schemes with flexible phase sequence are obviously unfeasible in real urban traffic network. Apart from that, quite a few results are obtained from simulation environment, the validity deserves further verification in practical settings.

In this paper, a novel RL-based scheme is proposed for TST problem. The main contributions include:

1) A novel traffic signal timing scheme called PGA is proposed. It is built upon the advantage actor-critic (A2C) architecture, and it integrates the state-of-the-art RL techniques proximal policy optimization (PPO) [14] and generalized advantage estimation (GAE) [15].
2) A feasible and secure signal timing scheme is derived according to PGA. Particularly, complexity of the model's state space is significantly reduced and the proposed TST scheme is with the fixed phase sequence, the alteration of successive phase durations is mild, which is preferable in field applications.
3) A new kind of reward function is proposed, and it significantly increases the convergence rate of the proposed scheme. Experimental results also indicate the traffic performance are remarkably improved by utilizing such reward function.

Comprehensive experiments are conducted based on real field-data, the results indicate that the traffic performance such as average queue length, average travel time and average vehicle delay are remarkably improved compared with the existing RL-based methods.

The rest of the paper is organized as follows: the related work on RL-embedded traffic signal timing is reviewed in Section 2. Background of reinforcement learning is briefly introduced in Section 3. Formulations of RL-based TST schme and the essential ingredients are presented in Section 4. In Section 5, we specify the proposed TST model and the relevant RL algorithm. In Section 6, simulation results and discussions are provided. The paper is concluded in Section 7.

2. Related Work

As a typical kind of machine learning method, RL has been adopted in adaptive traffic timing control since 1990s [16]. Model-free RL makes no assumptions about the model of environment and formulates signal timing as a sequential decision-making problem. It collects traffic information (such as traffic density, queue length and average velocity) as the states, generates appropriate actions to adjust the traffic light scheme, and improves the policy by traffic performance around the intersections. By this means, it achieves real-time adaptive control thoroughly. Comprehensive reviews are shown in [17], [18].

According to the type of embedded RL algorithm, RL-based TST approaches can be categorized into three classes: value function-based, policy gradient-based and actor-critic method. Value function-based algorithms such as Q-learning [19] and SARSA [6] optimize their behavior indirectly based on the approximation of certain optimal value functions. Most of early TST models prefer tabular Q-learning [20] or SARSA [9]. Designing schemes of state representation, action selection methods, reward definitions of tabular RL-TST model are investigated and compared in [21]. Most of them are coarse grained because the number of state-action pairs is limited. For example, the value of elapsed green time is converted from 0 to 50 seconds to a scaled integer value from 0 to 9 in [9]. Different relative queue size among four lanes are mapped into 24 integers to distinguish the states in [22]. Large amounts of information are wasted by this kind of design. However, if the fine-grained schemes are chosen, the capacity of Q-value tabular can be really large.

In order to alleviate the dimension curse of tabular RL and enhance its adaptability in large scale traffic environment, various value-based DRL methods are employed in traffic light control approaches [23]–[25], and turn out to be effective compensation to the tabular RL. The Q-value function is approximated by various deep neural networks in these articles, such as deep stacked autoencoders [23] and convolutional neural network [25]. Therefore, sophisticated RL elements such as state, action and reward can be used to enhance the method's effectivity.

However, value function-based RL inevitably requires massive computations in complex traffic scenarios, because the complexity of value function approximation hinges on the state space and action space. As an less commonly used alternative in TST, policy gradient-based method directly maps its observation to actions and updates the relevant parameters to achieve optimal policy. Calculation complexity is significantly reduced by this mechanism. [26] builds the deep policy-gradient and deep Q-learning traffic control models, simulation results indicate that they can both find stable control policies.

Actor-critic method [6] is the composite of policy gradient-based and value function-based algorithms, where the actor improves the policy via policy gradient methods, and the critic evaluates the policy by estimating certain value functions. [27] establishes discrete and

continuous actor-critic traffic signal controllers in recurrent congested traffic network, and compares different function approximators. [28] proposes a deep deterministic policy gradient(DDPG) based TST scheme. Continuous actions, variable phases sequence and cycle time are both adopted in it.

Superiority of RL-based TST approaches is fully displayed in these works, but the design of traffic observation, action schemes and reward function are still lack of safety and feasibility. For example, most of action representations are designed as index of next activated traffic phase [13], [26], [29], which means the sequence of traffic phase is out-of-order, and that is not safe in field applications. In addition, the reward in [24] is linear weighted sum of multiple factors, including queue length, vehicle delay, waiting time and total travel time. It's unnecessary to take too many elements into account. According to [30], some of these traffic metrics are relevant or equivalent, and there is no effective method to tune the weights for each component. Therefore, more efficient RL algorithm and more reasonable RL element schemes for TST are explored in this article.

3. Background of Reinforcement Learning

As mentioned in [6], RL enables an agent to interact with the environment and learn to map the situations into actions so as to maximize the expected total reward in a trial-and-error manner.

As illustrated in **Fig. 1**, at each step t, the agent acquires the current state s_t that reflects the real-time information of the environment. RL enables an agent to interact with the environment and learn to map the situations into actions so as to maximize the expected total reward in a trial-and-error manner. According to policy $\pi_\theta(a_t \mid s_t)$, the agent conducts action a_t and receives reward r_t from the environment. As a result of action a_t, the environment yields and transmits a new state s_{t+1} to the agent. Thus, the procedure $(s_1, a_1, r_1, \cdots, s_t, a_t, r_t)$ forms a trajectory of the agent. RL algorithms aim to provide the agent with optimal policy so as to achieve the learning target, i.e., to maximize certain cumulative rewards, by exploring possible trajectories and updating the policy.

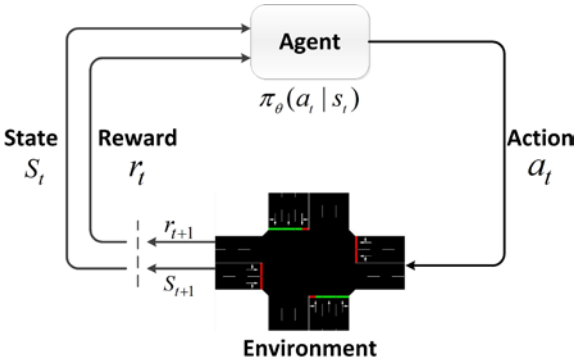


Fig. 1. Reinforcement learning procedure

In the following context, R_t denotes the discounted accumulated return from step t:

$$R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k} \tag{1}$$

where $\gamma \in [0,1]$ is the discounted factor, $\pi_\theta(a \mid s)$ denotes the policy parameterized by θ which essentially represents the action probability distribution under certain state:

$$\pi_\theta(a \mid s) = p(a = a_t \mid s = s_t) \tag{2}$$

Moreover, under policy π_θ, the probability of trajectory τ is denoted by $p(\tau;\theta)$. In the considered set-ups, the learning objective for the agent is to maximize the return by training the policy:

$$arg\max_\theta U(\theta) = \sum_\tau p(\tau;\theta) R_t \tag{3}$$

The estimation of the policy gradient can be computed and plugged into the gradient ascent algorithm [14]. For instance, in the vanilla policy-based method [31], it optimizes policy parameters θ in the direction of $\nabla_\theta U(\theta)$,

$$\nabla_\theta U(\theta) = \hat{\mathbb{E}}[\nabla_\theta log \pi_\theta(a|s) R_t] \tag{4}$$

where the empirical expectation $\hat{\mathbb{E}}[\cdots]$can be practically replaced by the average over a batch of real samples.

However, the accumulated return R_t in (4) is with high variance in general, which may decrease sample efficiency and result in unsatisfactory convergence result. To overcome this problem, the well-known actor-critic algorithm, called the Advantage Actor-Critic (A2C) [32], is illustrated as follows.

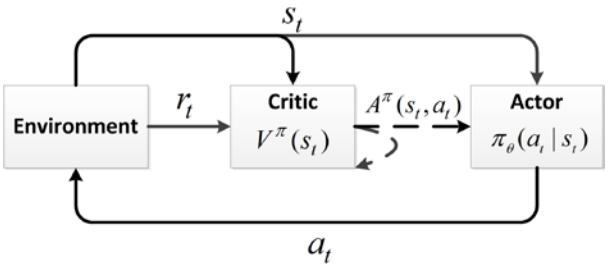


Fig. 2. Framework of Advantage Actor-Critic algorithm

Fig. 2 exhibits the architecture and basic components of A2C. The essential idea of A2C is to replace the accumulated return R_t in (4) with the advantage function $A^{\pi,\gamma}(s_t, a_t)$ to achieve the policy gradient,

$$A^{\pi,\gamma}(s_t, a_t) = Q^{\pi,\gamma}(s_t, a_t) - V^{\pi,\gamma}(s_t) \tag{5}$$

where $Q^{\pi,\gamma}(s_t, a_t)$ is the discounted state-action value function that represents the expected return by choosing action a_t under state s_t according to policy $\pi(a_t \mid s_t)$, and $V^{\pi,\gamma}(s_t)$ is the discounted state value function that denotes the expected return that initiating from state s_t according to policy θ, no matter what action is conducted.

$$\begin{aligned} Q^{\pi,\gamma}(s_t, a_t) &= \hat{\mathrm{E}}_{s_{t+1}, a_{t+1}, \ldots}[\sum_{k=0}^{T} \gamma^k r_{t+k}], \\ V^{\pi,\gamma}(s_t) &= \hat{\mathrm{E}}_{a_t, s_{t+1}, a_{t+1}, \ldots}[\sum_{k=0}^{T} \gamma^k r_{t+k}] \end{aligned} \tag{6}$$

Under (6), $V^{\pi,\gamma}(s_t)$ in (5) can be regarded as an average return or a baseline function, and $A^{\pi,\gamma}(s_t, a_t)$ can be considered as the measure of whether the action a_t is better than any other action $a_t^{'}$. The choice of such advantage function $A^{\pi,\gamma}(s_t, a_t)$ will lead to the estimation of

policy gradient with much lower variance. However, in practice, the advantage function is not known a priori and it must be estimated. An ideal unbiased estimator $\hat{A}_t$ of $A^{\pi,\gamma}(s_t, a_t)$ takes the following form:

$$\hat{A}_t = r_t + \gamma V^{\pi,\gamma}(s_{t+1}) - V^{\pi,\gamma}(s_t) \tag{7}$$

Consequently, replacing the accumulated return R_t in (4) with the advantage function estimator (7), it yields

$$\nabla_\theta U(\theta) = \hat{\mathrm{E}}[\nabla_\theta \log \pi_\theta(a \mid s)\hat{A}_t] \tag{8}$$

(8) facilitates A2C algorithm to calculate the policy gradients, and effectively resolves the high trajectory variance problem.

4. RL-based Traffic Signal Timing Formulation

To formulate the concerned traffic signal timing problem w.r.t. the reinforcement learning model and algorithm, in this section, the basic ingredients, namely the state, the action and the reward, are carefully defined. Particularly, a new form of the state is provided to reduce the complexity caused by the mainstream high dimensional state representation. The actions are carefully designed as the moderate change of traffic phase durations without modifying the phase sequence, which will lead to a smooth and more practical timing scheme. Also, the new definition of reward improves the learning efficiency of the algorithm.

4.1 State Space Representation

The representation of the state in RL models directly effects the computational complexity and system performance. In TST problems, the state of the environment refers to the traffic information of certain traffic control areas, e.g. the intersection, the arterial road, the elevated road, etc. In recent years, the Discrete Traffic State Encoding (DTSE) technique [23] is commonly adopted to characterize the traffic information around the intersections [25], [29]. DTSE usually retrieves the traffic information from large sequences of traffic images by discretizing lanes approaching the intersection into different cells. For each cell, the vehicle information such as the vehicles' presence and speed are recorded as certain matrices. Normally, DTSE may provide comprehensive and detailed traffic information to the agent. However, some experimental results indicate that there is no significant improvement can be achieved by using high-resolution traffic state such as DTSE rather than the traditional standard traffic parameters such as occupancy, queue length, mean speed, etc. for the same TST model [33]. Moreover, high-resolution traffic state dramatically increases the storage requirements and computational complexity.

Instead of DTSE, we define the following two types of traffic states: s_i^n and s_i^v, in the proposed RL model, where $i = 1, 2, \cdots, P$, and P is the number of phases in a signal cycle. To be specific, for the intersection under investigation, let N_i be the number of lanes associated with the i th phase, $n_{i,j}$ and $v_{i,j}$ represent the number of vehicles and mean speed of vehicles in lane j within the i th phase, respectively. Then s_i^n and s_i^v are defined as follows:

$$s_i^n = \frac{1}{N_i}\sum_{j=1}^{N_i} n_{i,j} \tag{9}$$

$$s_i^v = \frac{1}{N_i}\sum_{j=1}^{N_i} v_{i,j} \tag{10}$$

It is evident that s_i^n denotes the average number of vehicles in lanes under the i th phase. It reflects the traffic flow distribution among different directions of the intersection. And s_i^v represents the average speed of vehicles in lanes associated with the i th phase, which indicates the occupancy degree (jammed or under-saturated) of the lanes at the intersection. By (9) and (10), define the state space $S \in \mathbb{R}^{2P}$ for a signal timing scheme with P phases, and state $s_t \in S$ can be expressed as follows:

$$s_t = [s_1^n, \cdots, s_P^n, s_1^v, \cdots, s_P^v] \tag{11}$$

4.2 Action Space Representation

For TST problems, the agent perceives the environment by collecting traffic information, then according to the policy, it selects an action from the available action set under current state to modify the signal timing scheme. In our proposed RL model, we choose the fixed phase sequence scheme by considering the fact that it is more convenient and trustworthy in field applications. Therefore, the actions are defined to alter the durations of phases rather than alter the phase sequence. Besides, since the large change of phase durations between two consecutive cycles may cause certain problem for drivers and pedestrians, we set a constraint on the action so that each phase duration may only alter with a fixed adjustment. The choice of its value is crucial for the model' s performance, so we conduct comparison experiments to explore the optimal value in Section 6.

Consider a signal timing plan containing P phases, define the action space $A \in \mathbb{R}^{2P+1}$, and action $a_t \in A$ can be expressed as follows:

$$a_t = [a_1^+, \cdots, a_P^+, a_0, a_1^-, \cdots a_P^-] \tag{12}$$

where a_i^+ denotes the action of increasing the length of i th phase by 5s, a_i^- denotes the action of decreasing the length of i th phase by 5s, and a_0 represents no change are taken for all phases duration, $i = 1, 2, \cdots, P$. In addition, all phase durations are bounded by certain predefined minimum and maximum green time $g_{\min}$ and $g_{\max}$, respectively. This kind of action representation can be utilized to any intersections with variant number of phases. It is obvious that the dimensions of both state and action spaces are linear to the number of phases in the timing plan.

4.3 Reward Function Representation

After conducting the action, the feedback of the environment to the agent is characterized by an immediate reward. The definition of the reward is closely related to the optimal objective and plays the important role in achieving policy optimization. In the TST scenarios, the travel delay and queue length are commonly used as the reward since these standard traffic indexes reflect the mobility and traffic efficiency directly. Based on theoretical deductions and simulation results, [30] illustrated that using queue length as the reward function equals to optimizing the travel time. Considering the fact that with common sensors, e.g., surveillance cameras, the queue length is easier to get than any other information such as travel time, so in the proposed RL model, we use queue length as the main component of the reward function.

Notice that in the definition of advantage function (5), the state value function $V^{\pi,\gamma}(s_t)$ is subtracted from the state-action value function $A^{\pi,\gamma}(s_t, a_t)$. As previously discussed, this fact

implies that the state value function can be considered as the baseline which leads to much lower variance estimate of the policy gradient. Inspired by this idea, a new kind of reward is proposed in our RL model:

$$r_t = -[q_t - \frac{1}{T}\sum_{k=1}^{T} q_k] \tag{13}$$

where T is number of steps in an episode, q_t is the current queue length, and q_k denotes the history queue length in the last episode. An episode is one complete play of the agent interacting with the environment, which is composed of certain number of steps in this article. The TST model is trained in iterations, and one iteration equals to one episode.

In equation (13), the average queue length of the last iteration is considered to be the baseline, and it is subtracted from the metrics of current time step. The underlying mechanism of this kind of reward is to encourage the agent to optimize the policy on the basis of last episode. In addition, the second term in (13) is a constant for each step during one episode, so it will improve learning efficiency without increasing number of variables to be determined.

5. The PGA Algorithm for Traffic Signal Timing

In the previous section, we formulate the RL scheme for TST with the explicit representations of the state, action and reward. In this section, we develop a new kind of RL-based learning algorithm to achieve effective and efficient. Particularly, the proximal policy optimization (PPO) and generalized advantage estimation (GAE) techniques are adopted in the algorithm, where the former technique improves sample efficiency and simplifies the implementation procedure in a reliable manner, and the latter provides an effective variance reduction scheme for policy gradients. And it's based on the A2C architecture. Thus the proposed algorithm is termed the PGA (PPO-GAE-A2C) algorithm. The general architecture of PGA is shown in **Fig. 3**.

5.1 Proximal Policy Optimization

PPO inherited the ideas of trust region policy optimization (TRPO) [34] by introducing the following maximization problem w.r.t. surrogated objective function:

$$\begin{aligned} &arg\max_{\theta} \hat{E}_t[\frac{\pi_\theta(a_t|s_t)}{\pi_{old\theta}(a_t|s_t)}\hat{A}_t] \\ &subject\ to\ \hat{E}_t[KL[\pi_{old\theta}(\cdot|s_t), \pi_\theta(\cdot|s_t)]] \le \delta \end{aligned} \tag{14}$$

where $\pi_\theta(a_t \mid s_t)$ denotes the current policy, while $\pi_{old\theta}(a_t \mid s_t)$ means the policy before update, i.e., the old policy. As in (14), trajectories under the old policy can be used to optimize the current policy as long as the average KL divergence between the distributions of their parameters is smaller than the constraint value δ. This mechanism indeed transforms the on-policy method into the off-policy one, and more importantly, enables the policy-based RL algorithm to reuse the samples. In addition, the monotonic improvement of the policy can be guaranteed by using the surrogate objective function in (14).

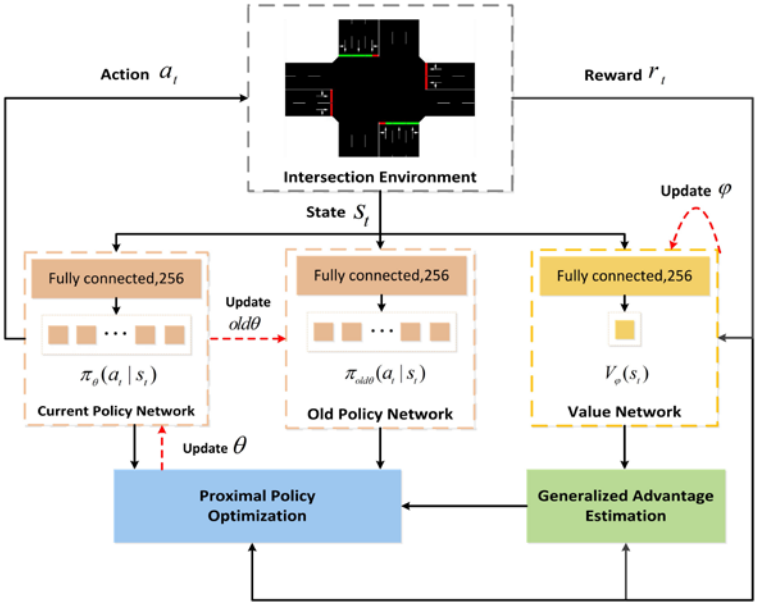


Fig. 3. Architecture of PGA algorithm

However, the high complexity of calculating the average KL divergence between two distributions reduces the training speed and limits the size of state and action spaces. Concerning these issues, a clipped surrogate objective is designed in PPO algorithm:

$$L^{PG}(\theta) = \hat{\mathbb{E}}_t[\min(r_t(\theta)\hat{A}_t, clip(r_t(\theta), 1-\varepsilon, 1+\varepsilon)\hat{A}_t)] \tag{15}$$

where $r_t(\theta)$ is the probability ratio:

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{old\theta}(a_t|s_t)} \tag{16}$$

The clip function in the second term of (15) restricts the ratio within boundary of $[1-\varepsilon, 1+\varepsilon]$, where ε is a hyperparameter (e.g., $\varepsilon = 0.2$). And the gradient estimation of certain value function can be obtained by differentiating the clipped surrogate objective (15). Using this simple clipping trick, PPO algorithm approximately enforces KL constraint without complicating the computation of the gradient. This implies that PPO attains TRPO's data efficiency and reliable performance, but it is much simpler to implement.

5.2 Generalized Advantage Estimation

The essential improvement of A2C algorithm to the classic gradient-based algorithm relies on the usage of certain value function such as the advantage function (5) instead of the accumulated return (1) to achieve the estimation of policy gradient as specified in (8). Consequently, the data efficiency is improved since the data are "remembered" by the value function in an efficient way to be reused, and the variance is also reduced. However, the theoretical form of the value function can not be directly used, any practical approximation of the value functions will introduce certain bias, which may cause the failure of algorithm convergence, or the convergence to a poor solution that is not even a local optimum. Therefore, the tradeoff between reducing the variance and introducing the bias must be carefully considered so as to achieve a RL model with better performance. Fortunately, GAE [15] technique provides an elegant solution to such problem.

To achieve an appropriate estimation of advantage function, the temporal difference (TD) method is utilized. Let V be the approximation of state value function $V^{\pi,\gamma}$, and define the TD

residual $\delta_t^V = r_t + \gamma V(s_{t+1}) - V(s_t)$. It is possible to show when the approximation V is exactly the state value function, i.e., $V = V^{\pi,\gamma}$,

$$\begin{aligned}\hat{\mathbb{E}}_{s_{t+1}}\left[\delta_t^{V^{\pi,\gamma}}\right] &= \hat{\mathbb{E}}_{s_{t+1}}[r_t + \gamma V^{\pi,\gamma}(s_{t+1}) - V^{\pi,\gamma}(s_t)] \\ &= Q^{\pi,\gamma}(s_t, a_t) - V^{\pi,\gamma}(s_t) \\ &= A^{\pi,\gamma}(s_t, a_t)\end{aligned} \tag{17}$$

(17) indicates that the TD-residual $\delta_t^{V^{\pi,\gamma}}$ is an unbiased estimator of the advantage function $A^{\pi,\gamma}$. This fact has been revealed as in (7). However, the utilization of the approximation V yields certain bias. Meanwhile, δ_t^V can be considered as a one-step estimator of advantage function, it can be possibly extended into multi-steps formulation as follows:

$$\begin{aligned}\hat{A}_t^{(1)} &= \delta_t^V \\ \hat{A}_t^{(2)} &= \delta_t^V + \gamma\delta_{t+1}^V \\ &\cdots \\ \hat{A}_t^{(k)} &= \textstyle\sum_{l=0}^{k-1} \gamma^l \delta_{t+1}^V\end{aligned} \tag{18}$$

Inspired by the TD(λ) algorithm [6], exponentially-weighted average of these k-steps estimators (18) forms the generalized advantage estimator (GAE):

$$\begin{aligned}\hat{A}_t^{GAE(\gamma,\lambda)} &\triangleq (1-\lambda)(\hat{A}_t^{(1)} + \lambda\hat{A}_t^{(2)} + \lambda^2\hat{A}_t^{(3)} + \cdots) \\ &= (1-\lambda)\left(\delta_t^{V^{\pi,\gamma}} + \lambda\left(\delta_t^{V^{\pi,\gamma}} + \gamma\delta_{t+1}^{V^{\pi,\gamma}}\right) + \cdots\right) \\ &= \textstyle\sum_{k=0}^{\infty}(\gamma\lambda)^k \delta_{t+k}^{V^{\pi,\gamma}}\end{aligned} \tag{19}$$

where $\lambda\epsilon[0,1]$ is the primary hyperparameter accounted for the tradeoff between bias and variance. Optimal ranges of γ and λ are verified to be [0.96,0.99] and [0.92,0.99] respectively through a bunch of experiments [15].

5.3 Model Structure and Algorithm

Combing the aforementioned PPO and GAE techniques, in this paper, a novel PGA learning algorithm is proposed in accordance with the specified TST problem. **Fig. 3** presents its general architechture.

This model contains the actor module and the critic module. Especially, the actor module consists two policy networks, which are referred to as the current policy network and old policy network, respectively. The inputs of these two networks are the same vectors containing state information, and their outputs are the action probability distributions representing the current policy π_θ and old policy $\pi_{old\theta}$, where θ and $old\theta$ are the relevant model parameters. In the critic module, the value network receives the state information and the output is the predicted state value function, where φ represents the model parameters. Artificial Neural Network (ANN) is used to realize these three networks. Consider the fact that the dimension of the state and action space of the proposed TST problem is not large, we only use a single fully-connected hidden layer in ANNs to avoid overfitting.

The objective function of current policy network is defined according to (15) (16) (19), while the old policy network doesn't need to be trained, it periodically duplicates the parameters from the current policy network. The learning objective of value network is defined as follows:

$$arg\min_{\varphi} \| \textstyle\sum_{k=0}^{B-t \bmod B} \gamma^k r_{t+k} - V_\varphi(s_t) \|^2 \tag{20}$$

where B is number of samples in a minibatch, $\sum_{k=0}^{B-t \bmod B} \gamma^k r_{t+k}$ is the discounted sum of rewards, $V_\varphi(s_t)$ is the predicted state value function. The predicted state value function will gradually approximate the true state value function by the optimization procedure. The general

architecture of PGA is quite straightforward and feasible to achieve policy evaluation and policy improvement.

Algorithm 1 PGA algorithm for traffic signal timing

Input: Parameters of PGA model in **Table 2**

Output: $\pi_\theta(a_t \mid s_t)$, $V_\varphi(s_t)$

Notations: N :maximum number of training episodes, EP_{LEN} : number of steps in each episode, B : number of samples in a minibatch

1. Initialize intersection environment and traffic signal timing scheme
2. Initialize parameters of policy networks and value network
3. **for** $episode = 1, N$ **do**:
4. Initialize environment and collect initial state s_t
5. **for** $timestep = 1, EP_{LEN}$ **do**:
6. Run old policy $\pi_{old\theta}(a_t \mid s_t)$ to sample trajectories
7. If number of trajectories $\geq B$:
8. Compute generalized advantage estimation according to (19)
9. Optimize current policy network with K epochs according to (15) (16)
10. Optimize value network with K epochs according to (20)
11. Duplicate parameters from current policy network to old policy network
12. **end for**
13. **end for**

Pseudo-code of PGA algorithm is given in Algorithm 1, the basic procedures are summarized as follows. In the beginning, traffic timing scheme and network parameters are randomly initialized. The initial state from the intersection is collected as the input of the old policy network in order to sample different minibatch size trajectories. Then the generalized advantage estimation is performed according to output of value network and the cumulative rewards, meanwhile, the parameters of both current policy network and value network are optimized for K epochs in the gradient descent manner w.r.t. (15), (16) and (20). Consequently, the parameters of current policy network are duplicated to the old policy network.

6. Experiment

6.1 Experiment Settings and Datasets

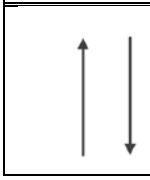
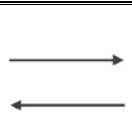
Experiments are conducted on traffic simulation platform SUMO (version 0.32.0). Traffic signal timing algorithms are developed on Python integrated development environment PyCharm (version 2018.2.1). The interactions between SUMO and the proposed algorithm are conducted through a Python module TraCI (Traffic Control Interface) in SUMO. TraCI retrieves the traffic information from the simulated environment and transfers them to the algorithm, it also performs signal timing schemes according to results provided by the TST models. All simulations were executed on a Windows desktop computer with an Intel CPU (i7-4720HQ, @2.60GHz), 16 GB RAM and a GeForce GTX 960M GPU.

In the experiments, TST models are built upon a real-world intersection of Shanyin Road and Shixin Road, Hangzhou, China. There are 20 lanes at the intersection, and its satellite map is shown in **Fig. 4**. Actual phase sequence of this intersection is illustrated in **Table 1**. In

practice, every phase is followed by a 3 seconds yellow light duration (including the red light clearance).

Fig. 4. Satellite map of intersection of Shanyin Road and Shixin Road in Hangzhou

Table 1. Phase sequence of signal timing scheme

Phase1	Phase2	Phase3	Phase3
			

Field traffic data used in simulations are collected from surveillance cameras at the intersection. **Fig. 5** illustrates the traffic flow data during July23-27, 2018, which are sampled every 5 minutes. Notice that the data in the blue and red boxes represent traffic status in regular periods (10:30-13:30) and rush hours (15:00-18:00) respectively. To test the performance and adaptability of the proposed signal timing strategy, experiments are conducted with respect to both regular periods and rush hours. The traffic flow data to initialize the simulation and to achieve the optimized timing schemes are those of July 25, 2018, while the data for testing are those of July 24, 2018.

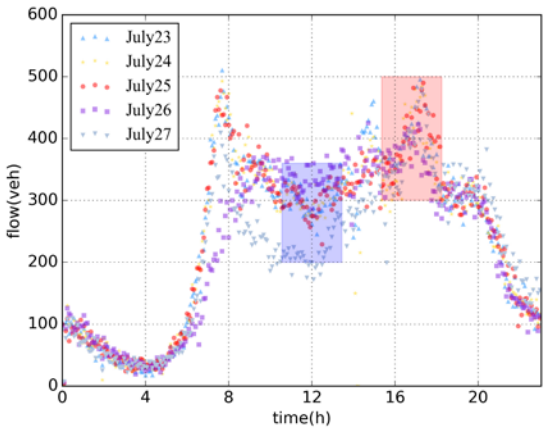


Fig. 5. Traffic flow of intersection of Shanyin Road and Shixin Road, July 23-27, 2018. The blue and red boxes contain traffic data in medium traffic density (10:30-13:30) and high traffic density (15:00-18:00) scenarios respectively

6.2 Evaluation Methods and Metrics

To validate the performance of PGA traffic signal timing scheme, comparative experiments are conducted according to the following state-of-art and classical RL-based TST methods.

- 3DQN[25]: Dueling double deep Q-network TST model adopts DTSE as traffic state representation, and its available actions indicate time intervals to be allocated for the next phase.
- nNQ[35]: Asynchronous n-step Q-learning TST model uses a vector composed of four traffic statistics for state representation, and its actions represent combinations of green signals for some non-conflicting movements in the following phase.
- DDQN[10]: Double deep Q-network TST model uses DTSE as well, and actions in this model are designed to determine the following signal phase.

To verify the functionality of PPO and GAE techniques, two simplified models, namely the PA model and GA model, are derived by removing GAE and PPO from PGA respectively. Also, to verify that the TST model with simplified state representation shows no performance degradation than those with DTSE, a model called PGA-DTSE is also developed, whose main components are identical with PGA except the state.

For the concerned traffic signal timing problem, the primary objective is to improve traffic efficiency by reducing queue length. So from field-applications perspective, the average queue length, average travel time and average vehicle delay are all chosen to be the evaluation metrics.

Parameters of all models are tuned separately to achieve the best performance, which are listed in **Table 2-3** respectively. In all experiments, one episode consists of 64 consecutive signal cycles (steps). As shown in **Table 2-3**, TST models need different numbers of training episodes to converge, PGA converges many times faster than nNQ and DDQN. PGA-DTSE needs twice the number of training iterations than PGA, and it takes longer time to execute a step, because the size of its state is dozens of times larger than the latter's. This fact explicitly indicates the proposed state can significantly reduce the training time.

Table 2. Parameters of PGA, PA, GA and PGA-DTSE traffic signal timing model

Parameter	PGA	PA	GA	PGA-DTSE
Discount factor γ	0.99	0.99	0.99	0.99
Actor learning rate	0.0001	0.0001	0.0001	0.0001
Critic learning rate	0.0002	0.0002	0.0002	0.0002
GAE parameter λ	0.96	--	0.96	0.96
Clipping Parameter ε	0.2	0.2	--	0.2
Minibatch size B	32	32	32	32
Convergence episode	100	150	150	200
Number of epochs per episode	64	64	64	64

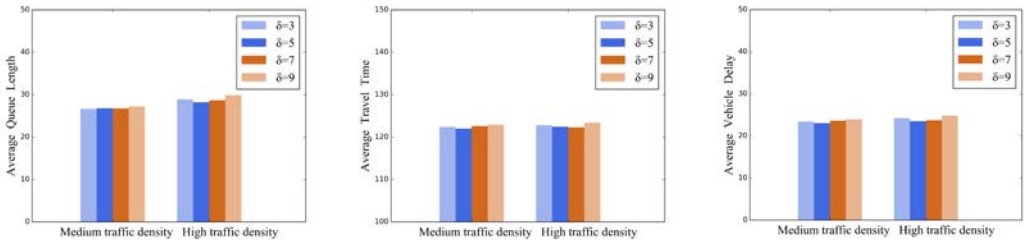
According to the definition of PGA's action in Section 4.2, during each step, duraion of one of the phases maybe shorten or prolonged by a fixed range, and that value is important to the model's performance. We record this value as δ, and comparison experiments are conducted to search the optimal solution. In real intersections, after the corresponding traffic light turning to green, it takes about 2-3 seconds for a vehicle to pass the stop line. So it makes no sense if δ is smaller than 2-3 seconds. At the same time, it's not secure to apply the TST scheme in

field application if δ is too large, the adjustment of traffic light timing should be smooth. So we choose 3, 5, 7, and 9 as candidate value for δ. Results of experiments in medium and high traffic density scenarios are shown in **Fig. 6**. Data in the figures are average values of 15 independent testing experiments.

Table 3. Parameters of 3DQN, nNQ and DDQN traffic signal timing model

Parameter	3DQN	nNQ	DDQN
Discount factor γ	0.99	0.99	0.90
Learning rate α	0.0001	0.0001	0.0001
Starting ε	1	1	1
Ending ε	0.01	0.0001	0.01
Minibatch size B	32	16	64
Replay memory size	10000	10000	10000
Convergence episode	150	1000	500
Number of epochs per episode	64	64	64

There are no explicit differences when δ is 3, 5, and 7. It demonstrates that PGA algorithm is qualified to explore optimal TST scheme, and not sensitive to the value of phase duration adjustments in this range. But when $\delta = 9$, the performance deteriorates and obvious vibrations happens. This means 9 seconds is too large, and it's hard to search an optimal solution in this situation. It's not wise to set δ larger than 9. According to the above experiments, δ is set to 5 in all simulations of this paper.



(a) Average queue length (b) Average travel time (c) Average vehivle delay

Fig. 6. Performance comparisons of different value of phase duration adjustment (recorded as δ) in medium and high traffic density scenario

6.3 Results and Discussions

Results of comparative experiments in medium and high traffic density scenarios are shown in **Table 4-5** respectively. The results are obtained by running 200 episodes on the trained models, according to the testing data. All data presented are averaged over 15 independent runs, and numbers after '$\pm$' are standard deviations.

It can be seen that PGA performs better than all others from all aspects. The simplified forms PA and GA are slightly weaker than PGA, but significantly better than other comparative models. This fact is in accordance with the theoretical discussions provided in Section 5, and it indicates that both PPO and GAE technique are essential for the proposed model. Their joint effect ensures the improvement of performance and adaptability of the signal timing scheme.

Testing results of PGA-DTSE are inferior to PGA, PA and GA, but better than the others. Particularly, in high traffic density scenario, the standard deviations of its metrics are pretty large, and its average vehicle delay is 87.79% higher than PGA. These outcomes demonstrate that the state consists of primitive traffic information leads no performance degradation. And the model with DTSE has weak adaptability in intersections with massive traffic volume.

The 3DQN model exhibits better performance than DDQN and nNQ. But compared with it, PGA reduces 12.6%, 6.7% and 41.5% of average queue length, average travel time and average vehicle delay in medium traffic density, and 27.4%, 8.1% and 49.5% in high traffic density.

Table 4. Testing results of different TST models in medium traffic density scenario by 15 independent runs. Numbers after ' ± ' are standard deviations of metrics

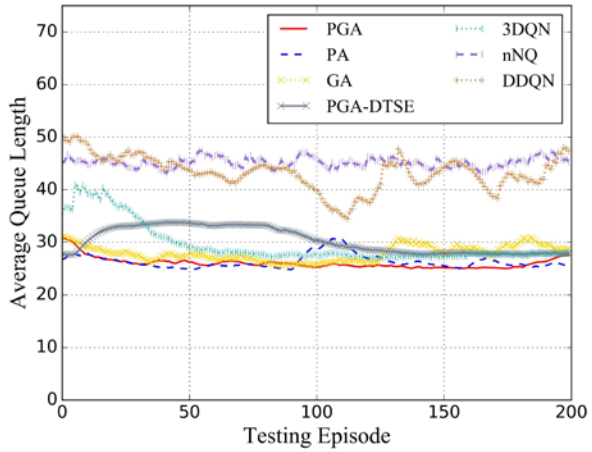
TST models	Average queue length(pcu)	Average travel time(s)	Average vehicle delay(s)
PGA	26.81 ± 1.37	122.00 ± 1.44	23.07 ± 1.41
PA	27.13 ± 1.91	122.59 ± 2.23	27.70 ± 1.98
GA	28.10 ± 1.67	123.45 ± 1.85	24.16 ± 1.50
PGA-DTSE	30.35 ± 2.44	126.76 ± 5.30	39.22 ± 1.73
3DQN	30.68 ± 4.95	130.86 ± 21.82	39.44 ± 21.84
DDQN	43.03 ± 3.52	138.52 ± 5.17	38.52 ± 4.48
nNQ	44.23 ± 1.05	147.47 ± 0.37	60.59 ± 0.37

Table 5. Testing results of different TST models in high traffic density scenario by 15 independent runs. Numbers after ' ± ' are standard deviations of metrics

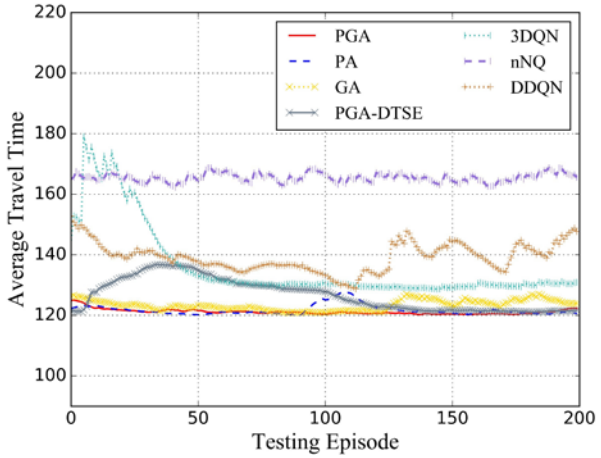
TST models	Average queue length(pcu)	Average travel time(s)	Average vehicle delay(s)
PGA	28.23 ± 0.71	122.45 ± 0.80	23.52 ± 0.76
PA	29.24 ± 0.91	123.47 ± 0.95	24.47 ± 0.88
GA	32.47 ± 2.42	128.15 ± 3.51	27.61 ± 2.48
PGA-DTSE	35.92 ± 8.49	130.01 ± 9.98	44.17 ± 9.98
3DQN	38.87 ± 5.86	133.28 ± 13.60	46.61 ± 13.60
DDQN	48.07 ± 10.66	146.36 ± 16.21	44.86 ± 14.47
nNQ	44.30 ± 0.32	163.90 ± 1.65	77.03 ± 1.65

It is worth mentioning that the adaptability of an algorithm naturally exhibits itself in high traffic density case. This is because it's challenging for the agent to adapt and learn smarter policy when the environment is continuously changing, i.e., there are more occurrences of the dramatic change of traffic flow in the high traffic density scenario. PGA shows prominent stability and generalization ability in high traffic density scenario. Compared to medium traffic density scenario, its performance metrics barely change and have smaller standard deviations than the others. In contrast, the performances of 3DQN and DDQN vibrate seriously. The trained nNQ model exhibits high level stability in both scenarios, but it has the longest average travel time and average vehicle delay in all experiments.

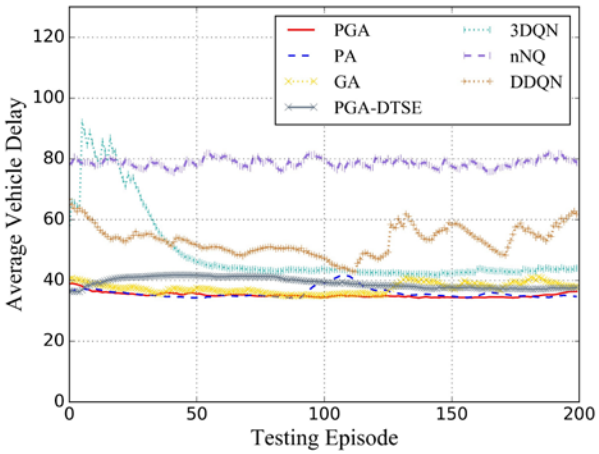
Overall comparisons of TST models under medium and high traffic density scenarios are shown in **Fig. 7-8** respectively. It can be seen that PGA achieves obvious advantage over all other models. PA and GA present closer metrics to the proposed model, but small vibrations can be found in their testing, especially in GA's high traffic volume testing. In medium traffic scenario, the performance of PGA-DTSE is closer to 3DQN. But it fluctuates badly when traffic volume around the intersection is large.



(a) Average queue length

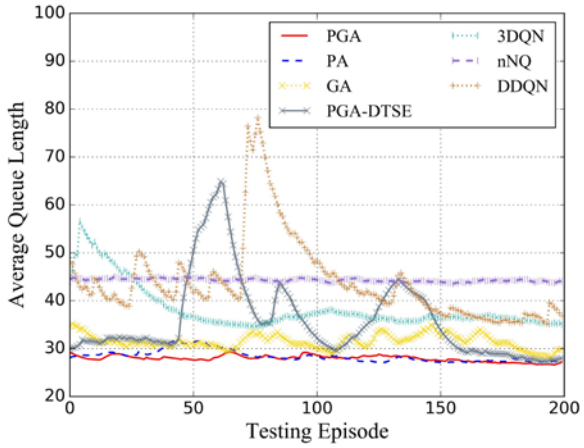


(b) Average travel time

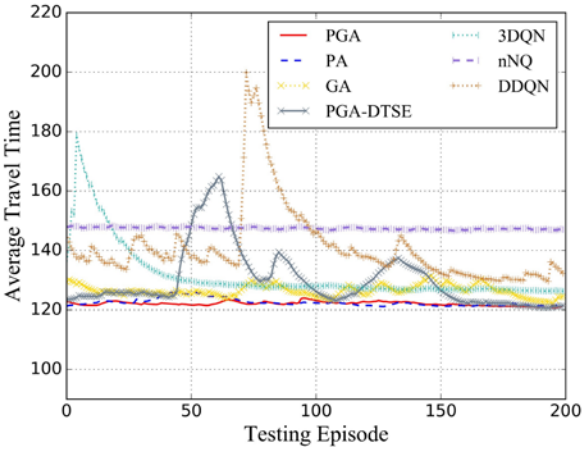


(c) Average vehicle delay

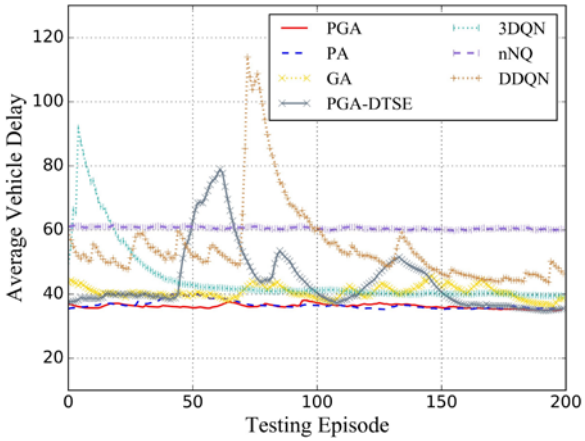
Fig. 7. Overall performance comparisons of adaptive signal timing models in medium traffic density scenario



(a) Average queue length



(b) Average travel time



(c)Average vehicle delay

Fig. 8. Overall performance comparisons of adaptive signal timing models in high traffic density scenario

What stands out about 3DQN is it needs about 50 episodes to re-adapt new environment in the beginning of every testing, then it converges to a relatively good and stable status. As shown in **Fig. 7-8**, performance of nNQ is consistent with data in **Table 4-5**, which indicates good stability and poor performance. Although DDQN outperforms nNQ in terms of average metrics, but what is shown in **Fig. 7-8** demonstrates that it has the worst adaptability and stability than the others.

To validate effectiveness of the new reward function proposed in the article, experiments are also conducted for the same PGA model with different reward functions. The results are provided in **Table 6**. In the “Reward definition” column, “Modified queue length” denotes the new kind of reward (13) employed in our model, “Queue length” denotes the direct usage of the average value of queue length at the intersection as the reward. Evaluation metrics of the trained models are recorded for both scenarios. It can be seen that the new reward function ensures the algorithm outperforms the one with traditional reward in all experiments. Especially, there is a significant improvement under high traffic density. Traffic light timing scheme with the modified reward function reduces 14.7% of average queue length, and 33.0% of average vehicle delay than the one with traditional reward function.

Table 6. Evaluation metrics of models with different reward functions, numbers after '± ' are standard deviations of metrics

Reward definition	Traffic density	Average queue length(pcu)	Average travel time(s)	Average vehicle delay(s)	Convergence episode
Modified queue length	Medium	26.81 ± 1.37	122.00 ± 1.44	23.07 ± 1.41	100
Queue length	Medium	28.13 ± 2.13	125.31 ± 1.89	27.65 ± 1.77	230
Modified queue length	High	28.23 ± 0.71	122.45 ± 0.80	23.52 ± 0.76	100
Queue length	High	33.13 ± 1.14	128.37 ± 1.91	35.11 ± 1.56	250

Another merit of the new reward function is regarding the convergence speed. According to the definition of the reward function in (13), it encourages the agent to optimize the policy on the basis of the effectiveness that achieved in the last episode. This mechanism efficiently decreases the model's training time. As exhibited in **Table 6**, it takes 100 episodes for the model with the new reward to achieve convergence, but it takes more than twice episodes to accomplish the training process with the traditional reward.

7. Conclusion

In this article, a RL-based adaptive traffic signal timing scheme PGA is proposed. The model is based on actor-critic architecture and it contains certain advanced RL techniques such as PPO and GAE. These techniques theoretically improve the effectiveness and efficiency of the relevant learning algorithm. Particularly, PPO improves sample efficiency in a brief and reliable implementation way, and GAE effectively reduces the policy gradients variance.

Meanwhile, considering the practical requirement of traffic signal timing problem, a new kind of reward function is defined, and simplifications of the state and action space representations are also introduced. The resulting signal timing plan significantly improves the general performances of traffic control systems compared with the existing ones such as 3DQN and DDQN-based schemes, also, it satisfies the requirements of feasibility, safety and adaptability in field applications. The test results are provided through real field traffic data-based simulations. To further refine the aforementioned scheme, pedestrians and non-motorized vehicles will be taken into consideration.

References

[1] G. Shen and Y. Yang, "A dynamic signal coordination control method for urban arterial roads and its application," *Front. Inf. Technol. Electron. Eng*., vol. 17, no. 9, pp. 907–918, 2016. Article (CrossRef Link)

[2] X. Kong et al., "Mobile Edge Cooperation Optimization for Wearable Internet of Things: A Network Representation-based Framework," *IEEE Trans. Ind. Informatics*, pp. 1-1, 2020. Article (CrossRef Link)

[3] X. Kong, J. Cao, H. Wu, and C. H. (Robert) Hsu, "Mobile Crowdsourcing and Pervasive Computing for Smart Cities," *Pervasive Mob. Comput*., vol. 61, 2020. Article (CrossRef Link)

[4] F. Zhang, J. Bai, X. Li, C. Pei, and V. Havyarimana, "An ensemble cascading extremely randomized trees framework for short-term traffic flow prediction," *KSII Trans. Internet Inf. Syst*., vol. 13, no. 4, pp. 1975–1988, 2019. Article (CrossRef Link)

[5] G. Shen, L. Zhu, J. Lou, S. Shen, Z. Liu, and L. Tang, "Infrared Multi-Pedestrian Tracking in Vertical View via Siamese Convolution Network," *IEEE Access*, vol. 7, pp. 42718–42725, 2019. Article (CrossRef Link)

[6] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction, 2nd ed*., The MIT Press, Cambridge, 2018.

[7] M. Wiering, "Multi-agent Reinforcement Learning for Traffic Light Control," in *Proc. of the Seventeenth International Conference on Machine Learning*, pp. 1151–1158, 2000.

[8] B. Abdulhai, R. Pringle, and G. J. Karakoulas, "Reinforcement Learning for True Adaptive Traffic Signal Control," *J. Transp. Eng.*, vol. 129, no. 3, pp. 278–285, 2003. Article (CrossRef Link)

[9] J. Jin and X. Ma, "A group-based traffic signal control with adaptive learning ability," *Eng. Appl. Artif. Intell*., vol. 65, pp. 282–293, 2017. Article (CrossRef Link)

[10] J. Gao, Y. Shen, J. Liu, M. Ito, and N. Shiratori, "Adaptive Traffic Signal Control: Deep Reinforcement Learning Algorithm with Experience Replay and Target Network," *arXiv:1705.02755*, 2017. Article (CrossRef Link)
[11] N. Casas, "Deep Deterministic Policy Gradient for Urban Traffic Light Control," *arXiv:1703.09035*, 2017. Article (CrossRef Link)
[12] E. van der Pol and F. A. Oliehoek, "Coordinated deep reinforcement learners for traffic light control," in *Proc. of 30th Conf. Neural Inf. Process. Syst.*, no. Nips, p. 8, 2016.
[13] W. Genders and S. N. Razavi, "Using a Deep Reinforcement Learning Agent for Traffic Signal Control," *arXiv:1611.01142*, 2016. Article (CrossRef Link)
[14] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," *arXiv:1707.06347*, 2017. Article (CrossRef Link)
[15] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-Dimensional Continuous Control Using Generalized Advantage Estimation," *arXiv:1506.02438*, 2015. Article (CrossRef Link)
[16] T. L. Thorpe and C. W. Anderson, "Traffic Light Control Using SARSA with Three State Representations," *IBM Corp.*, 1996.
[17] K.-L. A. Yau, J. Qadir, H. L. Khoo, M. H. Ling, and P. Komisarczuk, "A Survey on Reinforcement Learning Models and Algorithms for Traffic Signal Control," *ACM Comput. Surv.*, vol. 50, no. 3, pp. 1–38, 2017. Article (CrossRef Link)
[18] P. Mannion, J. Duggan, and E. Howley, "An Experimental Review of Reinforcement Learning Algorithms for Adaptive Traffic Signal Control," *Auton. Road Transp. Support Syst.*, pp. 47–66, 2016. Article (CrossRef Link)
[19] C. J. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, pp. 279–292, 1992. Article (CrossRef Link)
[20] I. Arel, C. Liu, T. Urbanik, and A. G. Kohls, "Reinforcement learning-based multi-agent system for network traffic signal control," *IET Intell. Transp. Syst.*, vol. 4, no. 2, pp. 128–135, 2010. Article (CrossRef Link)
[21] S. El-Tantawy, B. Abdulhai, and H. Abdelgawad, "Design of reinforcement learning parameters for seamless application of adaptive traffic signal control," *J. Intell. Transp. Syst. Technol. Planning, Oper.*, vol. 18, no. 3, pp. 227–245, 2014, Article (CrossRef Link)
[22] M. Abdoos, N. Mozayani, and A. L. C. Bazzan, "Hierarchical control of traffic signals using Q-learning with tile coding," *Appl. Intell.*, vol. 40, no. 2, pp. 201–213, 2013. Article (CrossRef Link)
[23] L. Li, Y. Lv, and F. Y. Wang, "Traffic signal timing via deep reinforcement learning," *IEEE/CAA J. Autom. Sin.*, vol. 3, no. 3, pp. 247–254, 2016. Article (CrossRef Link)
[24] H. Wei, H. Yao, G. Zheng, and Z. Li, "IntelliLight: A reinforcement learning approach for intelligent traffic light control," in *Proc. of ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp. 2496–2505, 2018. Article (CrossRef Link)
[25] X. Liang, X. Du, G. Wang, and Z. Han, "A Deep Reinforcement Learning Network for Traffic Light Cycle Control," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1243–1253, 2019. Article (CrossRef Link)
[26] S. S. Mousavi, M. Schukat, and E. Howley, "Traffic light control using deep policy-gradient and value-function-based reinforcement learning," *IET Intell. Transp. Syst.*, vol. 11, no. 7, pp. 417–423, 2017. Article (CrossRef Link)
[27] M. Aslani, M. S. Mesgari, and M. Wiering, "Adaptive traffic signal control with actor-critic methods in a real-world traffic network with different traffic disruption events," *Transp. Res. Part C Emerg. Technol.*, vol. 85, pp. 732–752, 2017. Article (CrossRef Link)
[28] H. Pang and W. Gao, "Deep Deterministic Policy Gradient for Traffic Signal Control of Single Intersection," in *Proc. of the 31st Chinese Control and Decision Conference, CCDC 2019*, pp. 5861–5866, 2019. Article (CrossRef Link)
[29] C.-H. Wan and M.-C. Hwang, "Value-based deep reinforcement learning for adaptive isolated intersection signal control," *IET Intell. Transp. Syst.*, vol. 12, no. 9, pp. 1005–1010, 2018. Article (CrossRef Link)

[30] G. Zheng et al., "Diagnosing Reinforcement Learning for Traffic Signal Control," *arXiv:1905.04716*, 2019. Article (CrossRef Link)
[31] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Mach. Learn*., vol. 8, no. 3, pp. 229–256, 1992. Article (CrossRef Link)
[32] T. Degris et al., "Model-Free Reinforcement Learning with Continuous Action in Practice," in *Proc. of 2012 American Control Conference (ACC)*, pp. 2177–2182, 2012. Article (CrossRef Link)
[33] W. Genders and S. Razavi, "Evaluating reinforcement learning state representations for adaptive traffic signal control," *Procedia Comput. Sci*., vol. 130, pp. 26–33, 2018. Article (CrossRef Link)
[34] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust Region Policy Optimization," in *Proc. of the 32nd International Conference on Machine Learning*, vol. 37, pp. 1889–1897, 2015.
[35] W. Genders and S. Razavi, "Asynchronous n-step Q-learning adaptive traffic signal control," *J. Intell. Transp. Syst. Technol. Planning, Oper*., vol. 23, no. 4, pp. 319–331, 2019, Article (CrossRef Link)

SI SHEN received the B.S. degree from Luoyang Normal University, Luoyang, China, in 2011, the M.S. degree from Criminal Investigation Police University of China, Shenyang, China, in 2014. She is currently pursuing the Ph.D. degree with the College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China. Since 2014, she has been with Zhejiang Police College, Hangzhou, China. Her current research interests include artificial intelligence, intelligent transportation and computational social science.

GUOJIANG SHEN received the B.S. degree in control theory and control engineering and the Ph.D. degree in control science and engineering from Zhejiang University, Hangzhou, China, in 1999 and 2004, respectively. He is currently a Professor with the College of Computer Science and Technology, Zhejiang University of Technology. His current research interests include artificial intelligence, theory, big data analytics, and intelligent transportation system.

YANG SHEN received the B.S. degree from Zhejiang University of Technology, Hangzhou, China, in 2019. He is currently pursuing the master's degree with the College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China. His current research interests include artificial intelligence and intelligent transportation.

4DUANYANG LIU received the B.Sc. degree in mechanical design and manufacture from Xiangtan University in 1997, and the M.S. degree in mechanical manufacturing and automation and the Ph.D. degree in computer science and technology from Zhejiang University, Hangzhou, China, in 2000 and 2003, respectively. He is currently an associate Professor with College of Computer Science and Technology, Zhejiang University of Technology. His current research interests include data mining, machine learning and intelligent transportation system.

XI YANG received the B.Eng. degree from Xi'an Jiaotong University, Xi'an, China, in 2004, the M.Eng. degree from Zhejiang University, Hangzhou, China, in 2007, and the Ph.D. degree from The Chinese University of Hong Kong, HKSAR, China, in 2011. Since 2012, he has been with the College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China. His current research interests include nonlinear control and optimization with applications in output regulation problem, multiagent systems, intelligent transportation systems, mathematical biology and image processing.

XIANGJIE KONG received the B.Sc. and Ph.D. degrees from Zhejiang University, Hangzhou, China. He is currently a Full Professor with College of Computer Science and Technology, Zhejiang University of Technology. Previously. Previously, he was an Associate Professor with the School of Software, Dalian University of Technology, China. He has published over 130 scientific papers in international journals and conferences (with over 100 indexed by ISI SCIE). His research interests include network science, mobile computing, and computational social science. He is a Senior Member of the IEEE and CCF and is a member of ACM.