IEEE INFOCOM 2021 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) | 978-1-6654-0443-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/INFOCOMWKSHPS51825.2021.9484451

Engineering A Large-Scale Traffic Signal Control: A Multi-Agent Reinforcement Learning Approach

Yue Chen*†, Changle Li*†‡, Wenwei Yue*†, Hehe Zhang*† and Guoqiang Mao*†
*State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an, Shaanxi, 710071 China
†Research Institute of Smart Transportation, Xidian University, Xi'an, Shaanxi, 710071 China
‡Email: clli@mail.xidian.edu.cn

***Abstract*—Reinforcement learning is of vital significance in machine learning and is also a promising approach for traffic signal control in urban road networks with assistance of deep neural networks. However, in a large scale urban network, the centralized reinforcement learning approach is beset with difficulties due to the extremely high dimension of joint action space. The multi-agent reinforcement learning (MARL) approach overcomes the high dimension problem by employing distributed local agents whose action space is much smaller. Even though, MARL approach introduces another issue that multiple agents interact with environment simultaneously causing its instability so that training each agent independently may not converge. This paper presents an actor-critic based decentralized MARL approach to control traffic signal which overcomes the shortcomings of both centralized RL approach and independent MARL approach. In particular, a distributed critic network is designed which overcomes the difficulty to train a large-scale neural network in centralized RL approach. Moreover, a difference reward method is proposed to evaluate the contribution of each agent, which accelerates the convergence of algorithm and makes agents optimize policy in a more accurate direction. The proposed MARL approach is compared against the fully independent approach and the centralized learning approach in a grid network. Simulation results demonstrate its effectiveness in terms of average travel speed, travel delay and queue length over other MARL algorithms.**

***Index Terms*—Multi-agent, deep reinforcement learning, traffic signal control, actor-critic.**

I. Introduction

With the growth of vehicle ownership, current transportation demand rises rapidly, especially in metropolises. However, the update speed of transportation infrastructure is slow. Moreover, the traffic signal in urban road networks is almost the fixed phase which can not fit the high transportation demand and causes congestion at intersections [1]. Adaptive traffic signal control (ATSC) can capture the dynamic variation rule of traffic flow and make a reasonable decision. Classical adaptive traffic signal control approaches are usually based on time gaps or time loss. *Time loss* based approaches work based on that phase prolongation can be triggered by the presence of vehicles with time loss to control a single intersection [2]. *Time gap* based approaches are common in Germany and work by prolonging traffic phases whenever a continuous stream of traffic is detected [3]. It switches to the next phase after detecting a sufficient time gap between successive vehicles. Several techniques such as fuzzy logic [4], evolutionary computation [5], [6] are also applied in adaptive traffic signal control.

In recent years, reinforcement learning (RL) approaches which are based on the framework of the Markov decision process (MDP) spring up in adaptive traffic signal control. It differs from traditional time loss or time gap based approaches which use a predesigned model. Rather it fits a parametric model whose inputs are collected from real traffic scenarios to learn the optimal control strategies by maximizing the reward function [7]. Traditional RL approaches that is represented by Q-learning usually use a simple model, leading the limited application in practice. However, the combination of deep neural networks and RL algorithms made great achievements in numerous complex tasks such as deep Q-learning (DQN) [8].

For the deep RL methods, there are three major methods: value based, policy based and value-policy mixed [9]. In value based methods, such as Q-learning, the action-state value function is fitted and its parameters are updated using step-wise experience. Tan *et al.* [10] adopted the bootstrapped Deep Q-Network (DQN) algorithm to induce exploration via an ensemble of behavior policies in traffic signal control. Wang *et al.* [11] developed a co-DQN method that is applied to traffic signal control and tested on various traffic flow scenarios of simulators. However, the update for DQN is based on the one-step temporal difference (TD), the non-stationary MDP transition of traffic scenario can not guarantee its convergence. For policy based methods, such as REINFORCE, the policy is parameterized by a deep neural network and updates its parameters using episode environment return by gradient ascent. Value-policy mixed method combines advantages of the aforementioned two methods, such as actor-critic (AC) method. In AC methods, the critic network evaluates the policy of each actor and guides them to optimize their policies. AC method had a low variance on gradient estimation so it converges fast than policy based methods [12]. A recent work [13] demonstrated that AC methods outperform Q-learning methods in traffic signal control.

Even though deep RL methods made its great achievements, but it is unpractical to train a centralized agent to control traffic signal in a large scale urban road network on account of extremely high-dimensional joint action space and joint state space which grow exponentially with the number of intersection. Under this circumstance, multi-agent

978-1-6654-0443-3/21/$31.00 ©2021 IEEE

reinforcement learning (MARL) approaches are utilized in traffic signal control. Early-stage MARL approaches use independent deep RL agents to control traffic signal. There is no communication among agents so that each agent only considers its own intersection state. Multiple agents simultaneously interact with the environment which causes the instability of the environment so that independent agent approaches usually have a poor convergence. Therefore, a few recent studies focus on a centralized learning and decentralized control MARL approach. Chu *et al.* [14] developed an advantage actor-critic method which utilized a centralized critic network and local actor to control traffic signal in a large scale road network. By improving the observability of each agent and considering the policies of other agents, this work implemented cooperative learning among multiple agents. In addition, the design of decentralized actor networks reduced the training difficulty. Nonetheless, utilizing a centralized critic network needs to collect all traffic measurements in the road network and transfer them to the processing center, which cause high latency and possibility of system breakdown once the communication outage. Besides, the centralized critic approach faces a credit assignment issue because it returns the same value to all agents which can not evaluate the contribution of each agent to global networks. In other words, the direction of policy improvement is not precise for each intelligent controller in traffic signal control.

To fill the gap, in this paper, we develop a decentralized critic network method for traffic signal control. To be specific, this study utilizes local actor networks and local critic networks. Each local agent broadcasts its state observation and receives state observations of other agents before learning. Then each local critic network approximates its own value function. For credit assignment, we utilize a difference reward based method to evaluate the contribution of each agent in cooperative game and accelerate the convergence. Finally, simulations are conducted based on Simulation of Urban Mobility (SUMO) to illustrate the effectiveness of proposed method[1]. Specifically, contributions of this paper are presented as follows:

- A distributed critic network is utilized to approximate the state value function in traffic signal control, which overcomes the difficulty to train a large-scale neural network and has a much smaller amount of data than centralized learning method. The fully decentralized design of actor and critic network also improves the robustness of traffic signal control systems.
- A difference reward based method is utilized in gradient estimation for updating actor network parameters, which can evaluate the contribution of each agent in cooperative game. The convergence speed of algorithm can be accelerated and policy can be optimized in a more accurate direction to receive a higher numerical reward.

[1]The demonstration of the simulated road network with different signal control strategies is available at https://github.com/albertcruzeyork/RL-for-traffic-signal-control

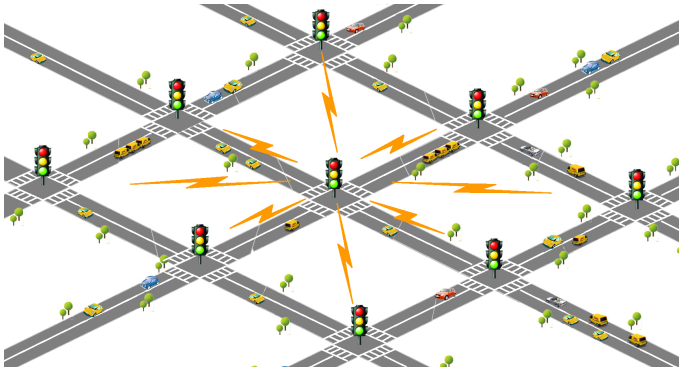


Fig. 1. A road network with traffic signal control agents.

- Simulations based on SUMO are conducted for different MARL algorithms. Reward over training steps and average travel speed under different traffic flows are also demonstrated, which illustrates the superiority of the proposed signal control strategy in average travel speed, travel delay and queue length.

II. Methodology

A. Policy Gradient

In a MDP, τ represents a series of state-action pairs $\{s_0, u_0, s_1, u_1, \cdots, s_h, u_h\}$, each trajectory τ receives a discount reward:

$$R(\tau) = \sum_{t=0}^{h} \gamma^t R(s_t, u_t), \tag{1}$$

where $\gamma \in [0, 1)$. For a parameterized policy π_θ, it hopes to maximize the discount reward. So the expectation of discount reward can be regraded as a target function to be optimized:

$$L(\theta) = E\left[\sum_{t=0}^{h} \gamma^t R(s_t, u_t) \mid \pi_\theta\right] = \sum_{\tau} P(\tau, \theta) R(\tau). \tag{2}$$

Optimization problem can be written as follows:

$$\max_{\theta} L(\theta) = \max_{\theta} \sum_{\tau} P(\tau, \theta) R(\tau). \tag{3}$$

To take the derivative of target function:

$$\begin{aligned}
\nabla_\theta L(\theta) &= \nabla_\theta \sum P(\tau, \theta) R(\tau) \\
&= \sum_{\tau} \nabla_\theta P(\tau, \theta) R(\tau) \\
&= \sum_{\tau} \frac{P(\tau, \theta)}{P(\tau, \theta)} \nabla_\theta P(\tau, \theta) R(\tau) \\
&= \sum_{\tau} P(\tau, \theta) \frac{\nabla_\theta P(\tau, \theta) R(\tau)}{P(\tau, \theta)}
\end{aligned}$$

$$= \sum_{\tau} P(\tau,\theta)\nabla_\theta \log P(\tau,\theta)R(\tau). \quad (4)$$

Eq. (4) can not be calculated directly, so Monte Carlo method is utilized to sample $R(\tau)$ from M trajectories and empirical average is used to estimate the gradient:

$$\nabla_\theta L(\theta) \approx \hat{g} = \frac{1}{m}\sum_{i=1}^{M} \nabla_\theta \log P(\tau,\theta)R(\tau^{(i)}). \quad (5)$$

Likelihood $P(\tau,\theta)$ in Eq. (5) can be represented as:

$$P(\tau^{(i)},\theta) = \prod_{t=0}^{h} P(s_{t+1}^{(i)} \mid s_t^{(i)}, u_t^{(i)}) \cdot \pi_\theta(u_t^{(i)} \mid s_t^{(i)}), \quad (6)$$

where $P(s_{t+1}^{(i)} \mid s_t^{(i)}, u_t^{(i)})$ represents the dynamics of system. So the gradient of $\log P(\tau,\theta)$ can be represented as follow:

$$\begin{aligned}\nabla_\theta \log P(\tau^{(i)},\theta) &= \nabla_\theta \log \prod_{t=0}^{h} P(s_{t+1}^{(i)} \mid s_t^{(i)}, u_t^{(i)}) \cdot \pi_\theta(u_t^{(i)} \mid s_t^{(i)}) \\ &= \nabla_\theta [\sum_{t=0}^{h} \log P(s_{t+1}^{(i)} \mid s_t^{(i)}, u_t^{(i)}) \\ &\quad + \sum_{t=0}^{h} \log \pi_\theta(u_t^{(i)} \mid s_t^{(i)})] \\ &= \sum_{t=0}^{h} \nabla_\theta \log \pi_\theta(u_t^{(i)} \mid s_t^{(i)}). \end{aligned} \quad (7)$$

Eq. (5) comes to be:

$$\nabla_\theta L(\theta) \approx \hat{g} = \frac{1}{m}\sum_{i=1}^{M}\sum_{t=0}^{h} \nabla_\theta \log \pi_\theta(u_t^{(i)} \mid s_t^{(i)}) R(\tau^{(i)}). \quad (8)$$

B. Policy Gradient with Baseline

When introduce a bias item which is only relevant to current state into reward, the gradient come to be:

$$\nabla_\theta L(\theta) = \nabla_\theta \log P(\tau,\theta)(R(\tau^{(i)}) - b(s)). \quad (9)$$

Introducing item b is unbiased for gradient estimation, because:

$$\begin{aligned} E\left[\nabla_\theta \log P(\tau,\theta)\cdot b(s)\right] &= \sum_{\tau} P(\tau,\theta)\nabla_\theta \log P(\tau,\theta) b(s) \\ &= \sum_{\tau} P(\tau,\theta)\frac{\nabla_\theta \log P(\tau,\theta) b(s)}{P(\tau,\theta)} \\ &= \nabla_\theta b(s) = 0 \end{aligned}$$

An appropriate b can minimize the variance of gradient estimation.

C. Advantage Actor-Critic

Current action is irrelevant to past reward. Based on that, the reward item in Eq. (8) can be replaced by this type reward as follows:

$$R_t(s_t,u_t) = \sum_{\tau=t}^{h} \gamma^{\tau-t} R\left(s_t,u_t\right). \quad (10)$$

Using expectation type:

$$Q^\pi(s_t,u_t) = E\left[\sum_{\tau=t}^{h} \gamma^{\tau-t} R\left(s_t,u_t\right) \mid s_t,u_t\right], \quad (11)$$

and utilizing a baseline:

$$V^\pi(s) = E_{u_t}\left[Q_\pi(s_t,u_t) \mid s_t\right]. \quad (12)$$

Eq. (8) can be written as:

$$\hat{g} = \frac{1}{m}\sum_{i=1}^{M}\sum_{t=0}^{h} \nabla_\theta \log \pi_\theta(u_t^{(i)} \mid s_t^{(i)})(Q^\pi(s_t,u_t) - V^\pi(s)), \quad (13)$$

where $A_t = Q^\pi(s_t,u_t) - V^\pi(s)$ called advantage function. Furtherly, $Q^\pi(s_t,u_t)$ can be written as:

$$Q^\pi(s_t,u_t) = E_{s_{t+1},u_{t+1}}\left[R\left(s_t,u_t\right) + \gamma Q^\pi(s_{t+1},u_{t+1})\right]. \quad (14)$$

So, A_t is approximated by:

$$R\left(s_t,u_t\right) + \gamma V^\pi(s_{t+1}) - V^\pi(s_t). \quad (15)$$

Critic introduces a neural network regressor $V_\omega(s)$ to estimate $V^\pi(s)$.

D. Distributed Critic Network

In a n agents game, each agent's state value function actually relevant to the states of other agents. For example, the state of agent a value function should be:

$$V_a^\pi(s_1, s_2, \cdots, s_n), \quad (16)$$

where s_i represents the state of agent i. In independent A2C method, actually the state value function was wrongly estimated by $V_a^\pi(s_a)$. Our critic networks distribute in each signalized intersection and broadcast its current state to all other agents at each transition step. A neural network V_ω is utilized to approximate the function in Eq. (16) with the coded states of all agents as the input.

E. Difference Reward

In comparison to centralized critic method which only considers global rewards, each agent's advantage function in our proposal is:

$$A_a = Q_a^{\pi}(s_1, s_2, \cdots, s_n, u) - V_a^{\pi}(s_1, s_2, \cdots, s_n), \quad (17)$$

where a represents the agent's serial number. Eq. (17) can be written as a time difference type:

$$A_a = R\left(s_a^t, u_a^t\right) + \gamma V^{\pi}(s_1^{t+1}, s_2^{t+1}, \cdots, s_n^{t+1}) - V_a^{\pi}(s_1^t, s_2^t, \cdots, s_n^t), \quad (18)$$

where $y_t^a = R\left(s_a^t, u_a^t\right) + \gamma V^{\pi}(s_1^{t+1}, s_2^{t+1}, \cdots, s_n^{t+1})$ is called TD target. Algorithm 1 illustrates our proposed algorithm.

Algorithm 1 Multi-Agent Actor Critic

for each agent i **do**
  Initialise θ_{π}^i, ω_c^i;
end for
for each training episode e **do**
  Empty buffer;
  for $e_c = 1$ **to** $\frac{Batchsize}{n}$ **do**
    initialise s and $t = 0$ for each agent;
    while $s \neq terminal$ and $t < T$ do
      $t = t + 1$;
      for each agent i **do**
        Sample u_t from π_{θ};
        Get reward r_t and next state s_{t+1};
      end for
    Add episode to buffer;
  end for
  Collate episodes in buffer into single batch;
end for
for each agent i **do**
  for $t = 1$ to T **do**
    Calculate TD targets y_t^a using ω_c^i;
  end for
  for $t = 1$ **to** T **do**
    $\triangle V_t^i = y_t^a - V_a^{\pi}(s_1^t, s_2^t, \cdots, s_n^t)$;
    $\nabla\omega_c^i = \nabla\omega_c^i + \frac{\partial}{\partial\omega_c^i}(\triangle V_t^i)^2$;
  end for
  $\omega_c^i = \omega_c^i + \alpha\nabla\omega_c^i$;
  for $t = 1$ **to** T **do**
    Caculate A_i;
    $\nabla\theta_{\pi}^i = \nabla\theta_{\pi}^i + \frac{\partial}{\partial\theta_{\pi}^i}\log\pi_{\theta}^i(u_t \mid s_t)A_i$;
  end for
  $\theta_{\pi}^i = \theta_{\pi}^i + \alpha\nabla\theta_{\pi}^i$;
end for

III. Simulation Results

A. MDP Settings

Considering a T seconds simulation environment, it is necessary to define a switch time $\triangle t$ as the period between two RL actions. If it is too long, the traffic signal control will not adaptive enough. If it is too short, the security can not be guaranteed and communication latency will influence the performance of traffic signal control systems. A yellow phase is also enforced between each switch. In this paper, the switch time is set to be 3s and the yellow phase duration is set to be 2s. Planing horizon is 4000 steps.

1) Action Definition: Several definitions are optional such as phase switch [15], phase duration [13] and phase itself [16]. This paper follow the third definition. It has predefined phase for each intersection such as red-green combination or yellow-red combination. Each agent will chose a phase at each step.
2) State Definition: This paper define each agent's state by velocities, distance to intersections, edge vehicle number (for nearby vehicles) from each direction, local edge information, and traffic light phase. That is: $s = \{v, l, n, phase\}$. Each edge in this paper is 500m long and all the state information are measured in 100m long to intersection.
3) Reward Definition: In this paper, each distributed agent receives its local step reward which is defined as: $R = -delay - queue$ which is inspired by [16] and delay is the average delay time of all vehicles and $queue$ is the length of vehicles standstill cloth to intersection. This reward is highly correlated to state and action of the local agent in comparison to cumulative delay [15] and wave [13].

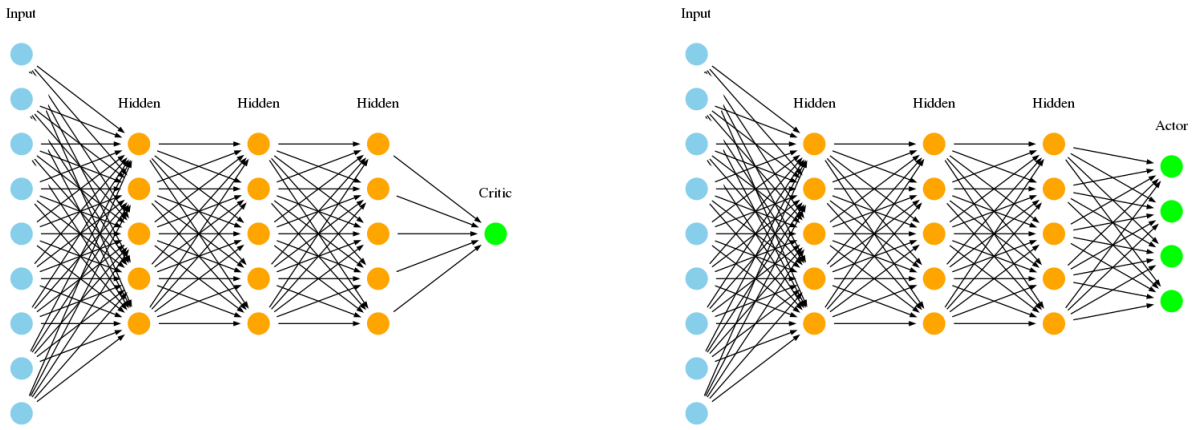
B. Neural Network Settings

In this paper, we utilize a fully connected network for actor network and critic network. Fig.2 illustrates the structure of our utilized neural network where joint state vectors (42*9 dimensions) are processed by fully connected layer as the input. Then critic network has three hidden layers with size of 256, 128 and 64. After each layer, $tanh$ activation is utilized. Last linear layer output the estimated value. Actor network use a softmax layer to output the probability distribution of actions. The gradient optimizer was chosen as stochastic gradient descent (SGD) with a learning rate 5e-5. In training, we utilize a mini-bacth method with a batchsize of 128.

C. Traffic Simulation Settings

MARL based traffic signal control is evaluated in SUMO and the utilized road network is a 3×3 grid like network as shown in Fig.1 under a high vehicle flow rate of 5000 vehs/h. The edge in the road network has two lanes with a speed limit 30m/s. To demonstrate the effectiveness and robustness of our proposed algorithm, we compare it to several algorithms which are utilized frequently in MARL studies including centralized critic method, independent A2C method. We train all algorithms up to 4M steps, which is around 1000 episodes with a episode horizon $T = 4000$ steps. For MDP parameters, we set $\gamma = 0.9$. Average travel speed which is defined as follow is utilized to evaluate the performance of each algorithm:

$$s = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m_i}\frac{v_j}{m_i}, \quad (19)$$



(a) Critic network. (b) Actor network.

Fig. 2. Architecture of critic network and actor network.

where n is the total simulation steps and m_i is the vehicle number in the road network at the step i. v_j is the speed of vehicle j.

D. Training Results

To evaluate the performance of each signal control method, average travel speeds of different traffic signal control methods are illustrated in Fig.3. A time gap based responsive method and fixed phase duration method were utilized as benchmarks. MAAC method has a best performance on the traffic flow of 3000 veh/h with an average travel speed of 64.04 km/h and outperforms other methods on the traffic flow of 1000 veh/h to 7000 veh/h. When the traffic flow is higher than 5000 veh/h, MAAC method start a downtrend and get close to the responsive method at the traffic flow of 8000 veh/h. As a contrast, centralized critic method and idependent A2C method start a sharp decline and even perform much worse than both benchmark methods When the traffic flow is higher than 5000 veh/h. On average, MAAC performs 59.2% better than fixed phase duration method, 29.3% better than responsive method, 12.8% better than centralized critic method and 35.4% better than independent A2C method (IPG was not taken into account because it does not converge).

Fig.4 plots the training result of each MARL algorithm with maximum episode reward, mean episode reward and minimum episode reward. The standard deviation of episode reward is illustrated in the figures. In Fig.4(a) and (b), training curves increase and then converge which shows the RL agents learn from cumulated experience and finally come to be optimum. But the converged reward in Fig.4(b) is approximately 260 numerical less than our MAAC approach showm in Fig.4(a) and in particular our MAAC approach has a faster convergence speed than centralized critic approach. In Fig.4(c), independent A2C takes much longer time to than MAAC and centralized critic that training curves converge and it can be found that the reward is much less than MAAC and centralized critic methods and the update process is noisy because the lack of effective communication among agents. In Fig.4(d), the training curves do not converge. Because the noisy gradient in

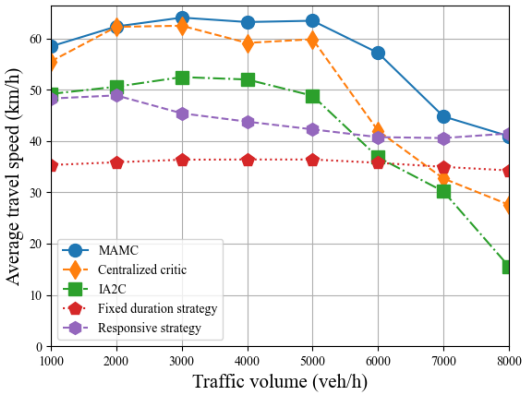
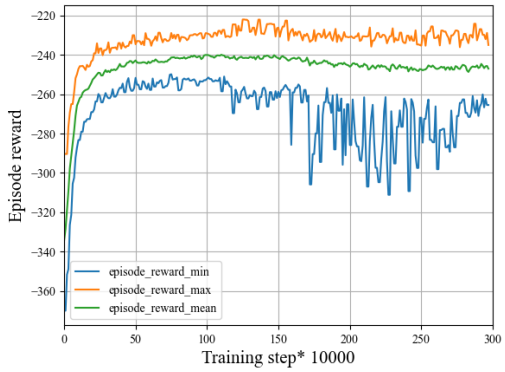


Fig. 3. Average travel speed in the road network.

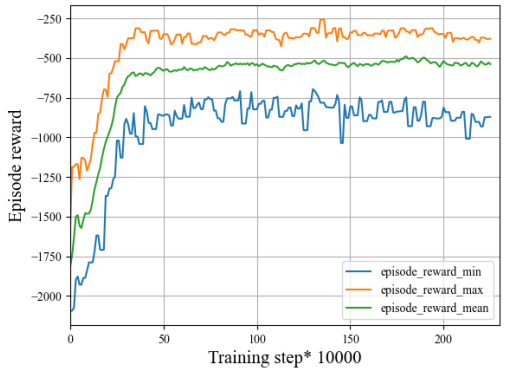
independent policy gradient method so that agents group can not be optimized in the right direction. The results show that our proposed approach has a faster convergence speed than centralized critic approach and idependent A2C approach. It is noteworthy that our proposed MAAC approach has a better performance than all other approaches utilized to be compared.

IV. Conclusion

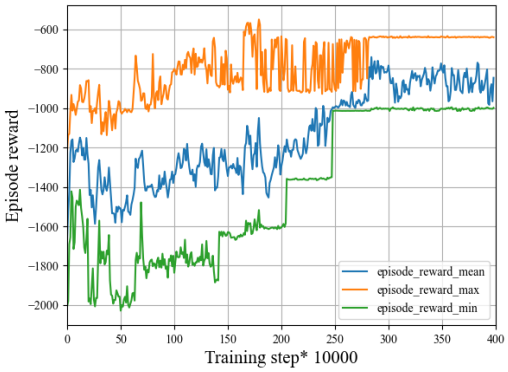
In this paper, a novel actor-critic based MARL algorithm for scaleable and robust traffic signal control was proposed. Firstly, we developed a decentralized critic network method for traffic signal control which avoid the difficulty to train a large-scale neural network in comparison to centralized learning method. The fully decentralized design of actor and critic network also improves the robustness of traffic signal control systems. Secondly, we utilized a difference reward method to solve the issue of credit assignment in multi-agent reinforcement learning which can evaluate the contribution of each agent in cooperative game. In experiment, MAAC had a highest mean episode reward of -240.11 and performed 59.2% better than fixed phase duration method, 29.3% better



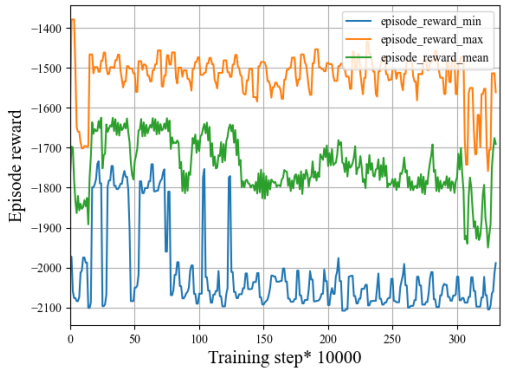
(a) MAAC.



(b) Centralized critic.



(c) Independent A2C.



(d) Independent PG.

Fig. 4. Training results of different MARL algorithms.

than responsive method, 12.8% better than centralized critic method and 35.4% better than independent A2C method with the evalution index of average travel speed.

Future works are still remaining for the MAAC algorithm which include: 1) experiment to verify the robustness of MAAC system when a number of agents break down; 2) utilize a recurrent neural to model the actor for better utilizing the history information.

V. Acknowledgment

This work was supported by the National Key R&D Program of China (2019YFB1600100), National Natural Science Foundation of China (U1801266), the Youth Innovation Team of Shaanxi Universities, and the Science and Technology Projects of Xi'an, China (201809170CX11JC12).

References

[1] C. Li, W. Yue, G. Mao and Z. Xu, "Congestion propagation based bottleneck identification in urban road networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 4827-4841, 2020.

[2] O. Robert and P. Wagner, "Delay-Time actuaed traffic signal control for an isolated intersection," *Transp. Res. Board*, pp. 1-6, 2011.

[3] S. Cho and R. Rao, "Coordinated ramp-metering control using a time-gap based traffic model," *in Proc. IEEE Veh. Technol. Conf.*, pp. 1–6, 2014.

[4] S. Chiu, "Adaptive traffic signal control using fuzzy logic," *in Proc. Intell. Veh. Symp.*, pp. 98-107, 1992.

[5] S. Darmoul, S. Elkosantini, A. Louati and L. B. Said, "Multi-agent immune networks to control interrupted flow at signalized intersections," *Transp. Res. C, Emerg. Technol.*, vol. 82, pp. 290-313, 2017.

[6] H. Ceylan and M. G. Bell, "Traffic signal timing optimisation based on genetic algorithm approach, including drivers' routing," *Transp. Res. B, Methodol.,* vol.38, no. 4, pp. 329-342, 2004.

[7] C. Watkins and P. Dayan, "Q-learning," *Mach. Learn.,* vol. 8, no. 3-4, pp. 279-292, 1992.

[8] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint*, arXiv:1312-5602, 2013.

[9] S. Mousavi, M. Schukat and E. Howley, "Deep reinforcement learning: An overview," *in Proc. SAI Intell. Syst. Conf.,* pp. 426-440, 2018.

[10] T. Tan, T. Chu and J. Wang, "Multi-agent bootstrapped deep Q-network for large-scale traffic signal control," *in Proc. IEEE Conf. Control Technol. Appl.*, pp. 358-365, 2020.

[11] X. Wang, L. Ke, Z. Qiao and X. Chai, "Large-scale traffic signal control using a novel multiagent reinforcement learning," *IEEE Trans. Cybern.*, vol. 51, no. 1, pp. 174-187, Jan. 2021.

[12] V. Konda and J. Tsitsiklis, "Actor-critic algorithms," *Adv. neural inf. proces. syst.*, pp. 1008-1014, 2000.

[13] M. Aslani, M. 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. C, Emerg. Technol.*, vol. 85, pp. 732-752, 2017.

[14] T. Chu, J. Wang, L. Codecà and Z. Li, "Multi-agent deep reinforcement learning for large-scale traffic signal control," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 1086-1095, Mar. 2020.

[15] S. El-Tantawy, B. Abdulhai and H. Abdelgawad, "Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (MARLIN-ATSC): methodology and large-scale application on downtown toronto," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1140-1150, 2013.

[16] P. LA and S. Bhatnagar, "Reinforcement learning with function approximation for traffic signal control," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 2, pp. 412-421, June 2011.