Emerging Cutting-Edge Applied Research and Development in Intelligent Traffic and Transportation Systems, M. Shafik (Ed.)

doi:10.3233/ATDE241191

Reinforcement Learning-Based Traffic Light Control Using a Clipped Double Deep Q-Learning Algorithm

Yanzhao JIA [a,b,1], Jianlin LU [a] and Daniel GÖRGES [a]

[a] *Yanzhao Jia, Jianlin Lu and Daniel Görges are with Institute of Electomobility, University of Kaiserslautern-Landau, 67663 Kaiserslautern, Germany. yjia@rhrk.uni-kl.de, jlu@rhrk.uni-kl.de, goerges@eit.uni-kl.de*

[b] *Yanzhao Jia is also with DENSO Automotive Germany GmbH, 41844 Wegberg, Germany. y.jia@eu.denso.com*

Abstract. This paper presents the results of intelligent traffic light management (TLM) which improves the traffic efficiency via optimally controlling the green lights' time interval and selecting the traffic phases. As the control problem is stochastic and difficult to be modeled accurately, model-free reinforcement learning (RL) is applied in this work. To stabilize the training process and mitigate the overestimation issue of conventional deep Q-learning based RL methods, we developed an RL algorithm with double deep Q-network (DQN) and a clipping function for the TLM problem with a discrete action space. The advantage of this clipped version of double DQN over other Q-learning-based algorithms is demonstrated in this work. Furthermore, the performance of RL-based TLM is compared with both fixed-time and adaptive rule-based TLM by using PTV Vissim which is a multi-modal traffic simulation software as the testing platform in this work.

Keywords. Traffic Light Management (TLM), Reinforcement Learning (RL), Clipped Double Deep Q-Learning (CDDQN)

1. INTRODUCTION

Traffic congestion at signalized intersections becomes an increasingly serious problem in modern cities. Traffic light management (TLM) plays a pivotal role in managing traffic flow at intersections. Two conventional methods for traffic signal control are fixed-time control [1] and flow-rate-based control [2]. These methods are cycle-based, relying mainly on statistical traffic information, which leads to sub-optimal performance in handling real-time traffic dynamics. In recent decades, numerous advanced methods [3], [4], [5], [6] have been developed for traffic signal management. With the growing availability of infrastructure sensors, these methods enable an automatic adaptation of traffic signal control according to the traffic status. Among these advanced methods, optimization-

[1] *Research work in this paper is done with a close collaboration between University of Kaiserslautern-Landau and DENSO Automotive Deutschland GmbH.

based [3], [7], [8] and machine learning (ML)-based [5], [9], [10] approaches are widely utilized.

A big challenge in developing optimization-based TLM is to have a precise model to accurately quantify the stochastic dynamics of the traffic system [11, 12]. Furthermore, even though a nonlinear mathematical model, which is representative of the traffic behavior, is built up in some previous work [3, 13], solving such optimization problem for real-time implementation is in general still challenging. With the development of artificial intelligence, intelligent TLM based on deep reinforcement learning (RL) has been proposed in recent works, among which the deep Q-learning network (DQN) stands out as the most frequently employed method due to its compatibility with discrete control tasks such as traffic signal phase selection [5, 9, 14], as well as the optimized decision variables including the extension or termination of traffic signal duration [10, 15]. Furthermore, other RL methodologies for continuous control tasks, such as deep deterministic policy gradients (DDPG), twin delayed DDPG (TD3), and soft actor-critic (SAC) have also been applied in traffic signal control systems [14,16,17], where the primary control variables involve the duration of a traffic signal phase within a fixed predefined sequence.

In our work, since the control variable is discrete, we focuses on the RL with Q-learning. Although it has been proven that the Q-learning-based algorithm converges to the true optimal value in the tabular case [18], when the deep Q network (DQN), which approximates the true Q value with a deep neural network, is applied, the convergence in general can not be guaranteed in the training process. One typical issue of DQN is overestimation bias [19], which is caused by using the function approximation and bootstrapping together. As an effective approach to mitigate the issue connected to value overestimation, double DQN has been proposed in [20]. Furthermore, a clipped double Q-learning variant TD3 shows its advantages in the setting of actor-critic RL in [21]. In this paper, we adapt the clipped double Q-learning which was originally proposed for solving a continuous control problem, to our traffic light management problem with discrete control variables.

The rest of this paper is organized as follows. The general concept and development environment for intelligent TLM are introduced in Section 2. Afterwards, the motivation of using the new clipped double Q-learning algorithm for intelligent TLM is explained in Section 3, as well as the performance comparison of different Q-learning algorithms. Furthermore, the advantages of the intelligent TLM is demonstrated in Section 4 via a comparison with other conventional TLM methods. Lastly, the conclusions and future work of this paper are given in Section 5.

2. PROBLEM FORMULATION OF INTELLIGENT TRAFFIC LIGHT MANAGEMENT

2.1. Development Environment in Simulation

In this work, the microscopic traffic simulation at a three-lane signalized intersection is built up with PTV Vissim, as illustrated in Fig. 1. The reinforcement learning (RL) algorithms are developed with Python, which has the access to control the traffic signals at the simulated intersection through the component object model (COM) interface provided by PTV Vissim. Two detectors are located at the entrance and exit of each lane respectively, through which the information about the number of vehicles that are driving in the same direction, as well as the number of vehicles that have passed through the intersection, can be obtained and sent to the RL-based controller. Eight traffic signal phases are used in this work, as shown in Fig. 2. To enable non-conflicting vehicular movements, the traffic lights are not allowed to become green for two or more phases at the same time. Different from the traditional fixed-time TLM, which often uses a pre-determined sequence and a fixed duration of the phases, RL-based TLM optimally chooses the phase and decides its duration based on the received information about the traffic status.

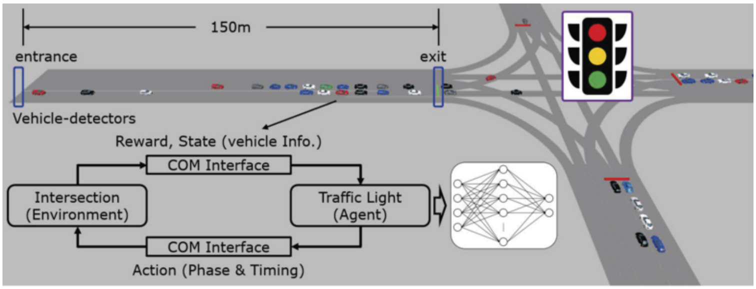

Figure 1. Platform of simulating the intelligent TLM

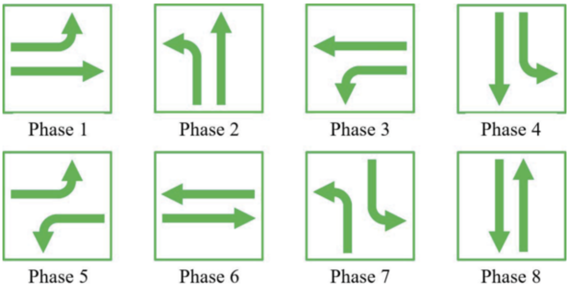

Figure 2. Design of either traffic phases

2.2. *Formulation of Reinforcement Learning for Traffic Light Management*

To design the TLM, the state, action and the reward function designed for RL algorithms in this work are explained in this subsection.

2.2.1. *State*

In this paper, the traffic state at the intersection consists of two components: instantaneous and average traffic information. The instantaneous information is represented as a vector with 8 elements: $N = [n_1, n_2, \ldots, n_8]$, corresponding to the number of vehicles driving in each phase at every time instant. The statistical traffic information is characterized by a vector with 4 elements $M = [m_w, m_s, m_e, m_n]$, indicating the average hourly traffic flow rates from 4 incoming directions. Hence, the total state space at time step t can be represented as a set of $\mathscr{S}_t = \{N_t, M_t\}$.

2.2.2. Action

To get higher flexibility of optimizing the traffic efficiency, the RL-based intelligent TLM designed in this work is not based on cycles, which means that the phases can be switched arbitrarily. For this purpose, the discrete decision variable Φ encoding the phases 1 to 8 is introduced. In contrast, with the cycle-based TLM the phases are switched periodically from the first to the last phase, providing no flexibility to adjust the phase's selection according to the dynamic traffic status. In addition, the duration of the green traffic light that is set for the selected phase controlled by another discrete variable Ψ. In this work, based on empirical testing experience, we designed 8 values for Ψ, i.e., $\Psi = \{6\,\text{sec}, 8\,\text{sec}, \ldots, 20\,\text{sec}\}$. As the selection and duration of the phases are optimized together, the dimension of discrete action space $\dim[\mathscr{A}_t]$ at time step t equals to $dim[\Phi_t] \cdot \dim[\Psi_t] = 64$.

2.2.3. Reward Function

The reward function is designed to guide the agent to learn a better control policy by providing an immediate feedback on the performance of the selected action at time step t. In our work, the main goal is to increase the traffic efficiency by encouraging more vehicles to pass through the intersection and simultaneously minimizing the waiting time of halting vehicles. Therefore, the reward function is designed as:

$$r_t = \omega_1 N_t^{\text{pass}} - \omega_2 T_t^{\text{wait}} \tag{1}$$

where N_t^{pass} is defined as the total number of vehicles that will drive through the intersection after one action a_t is taken at time t and before the next action a_t' is applied. The second item T_t^{wait} in (1) is the total waiting time of the vehicles from all driving directions between two actions a_t and a_t', as some vehicles have to halt when the green light is assigned to one specific phase. It needs to be pointed out that the time interval between taking a_t and a_t' is varying as it is decided by Ψ in this work. With our empirical experience, having the first item of N_t^{pass} in (1) is important. If r_t were only related to T_t^{wait}, RL would be encouraged to take the action with the smallest value of Ψ, because the observed T_t^{wait} within a shorter time interval tends to be also smaller, while our real goal is to reduce the average waiting time in a longer entire simulation period.

3. CLIPPED DOUBLE Q-LEARNING FOR INTELLIGENT TRAFFIC LIGHT MANAGEMENT

For the vanilla version of Q-learning-based RL, the parameterized state-action value function $Q_\theta(s,a)$ is learned by updating the parameter θ as

$$\theta_{t+1} = \theta_t + \alpha(y_t^{\text{tar}} - Q_{\theta_t}(s_t,a_t))\nabla_{\theta_t} Q_{\theta_t}(s_t,a_t) \tag{2}$$

where α is the learning rate and y_t^{tar} is named target which is defined as

$$y_t^{\text{tar}} = r_t + \gamma Q_{\theta_t}(s_t', \underset{a_t'}{\text{argmax}} Q_{\theta_t}(s_t',a_t')) \tag{3}$$

where r_t is the immediate reward by taking the a_t at state s_t, s_t' and a_t' are the state and action at the next time step respectively.

3.1. Motivation of Using Clipped Double Q-learning

As proven in [19, 22], assuming that we have an estimator μ_i which gives an unbiased estimate for the true expected value of a random variable X_i, i.e., $E[\mu_i] = E[X_i]$, the maximal estimator $\max_i \mu_i$ is an overestimation of $\max_i E[X_i]$, i.e., $E[\max_i(\mu_i)] \geq \max_i E[X_i]$. This indicates that the arg max operation in (3) would also result in an overestimation issue, even if the approximation error of using the network Q_θ has a mean of zero. Overestimation of the true value function can cause a serious issue for Q-learning-based RL, which uses deep neural networks for the function approximation. What makes the training process worse is that updating the parameterized Q function is done via bootstrapping, i.e., the Q value is in part used to update the Q value itself, due to which the overestimated value can be propagated through the Bellman equation. The detrimental impact of having a function overestimation on the performance of Q-learning-based RL is demonstrated via testing of games in [20].

To mitigate the overestimation issue, the double deep Q-network (DQN) has been proposed in [20]. The key idea of double DQN is to decouple the action selection and value evaluation steps by using two neural networks Q_θ and Q_{θ^-} with two sets of weights θ and θ^-. For each update, the greedy action is selected based on one network, while its state-action value is evaluated by another network . The new approach of calculating the target for updating θ is given as

$$y_t^{\text{tar}} = r_t + \gamma Q_{\theta^-}(s_t', \underset{a_t'}{\text{argmax}} Q_\theta(s_t', a_t')) \tag{4}$$

where the action a' is selected by the greedy policy based on the network Q_θ and the value of taking this action at state s' is evaluated by the other network Q_{θ^-}. Intuitively, the motivation of using two neural networks is that we hope that the approximation errors of two independent networks can be compensated by each other, assuming that the error of each network has a mean of zero.

The theoretical proof that using double estimators under specific conditions can change overestimation to underestimation of the true maximal expected value has been given in [19]. To further greedily reduce overestimation caused by function approximation, a clipped version of double DQN for actor-critic RL with continuous control variables is proposed in [21]. The clipping function takes the minimum value of two neural networks as

$$y_t^{\text{tar}} = r_t + \gamma \cdot \min_{j \in \{1,2\}} \left\{ Q_{\theta_j^-}(s_t', \underset{a'}{\text{argmax}} Q_{\theta_1}(s_t', a_t')) \right\} \tag{6}$$

The main purpose of using this clipping function is to ensure that the minimum value of two approximation functions is taken to make the target update.

In this paper, we extend the clipped version of double DNQ, which was originally developed for actor-critic RL to Q-learning-based RL, because the control variables for the application problem in this work are discrete. The pseudo-code of the clipped version double DQN algorithm with discrete control space is shown in Algorithm 1.

PseudoCode 1: Clipped Double Deep Q-Network for intelligent TLM with Discrete Control Variables

```
Initialize replay buffer $B$
Initialize policy networks $Q_{\theta_1}$, $Q_{\theta_2}$ with random parameters $\theta_1$, $\theta_2$
Initialize target networks $Q_{\theta_1^-}$, $Q_{\theta_2^-}$ with parameters $\theta_1^- = \theta_1$, $\theta_2^- = \theta_2$
Set exploration rate $\varepsilon$ and decay schedule, set discount factor $\gamma$, set learning
  rate $\alpha$, set soft update parameter $\tau$
for $episode = 1$ to $M$ do
    Initialize state $s_0$
    for $timestep\ t = 1$ to $T$ do
        Select action $a_t$ with $\varepsilon$-greedy policy and observe reward $r_t$ and next
          state $s_t'$
        Store transition $(s_t, a_t, r_t, s_t')$ in $B$
        Sample mini-batch of N transitions $(s_i, a_i, r_i, s_i')$ from $B$ randomly
        Compute target:
        $y_i = r_i + \gamma \min_{j \in \{1,2\}} Q_{\theta_j^-}(s_i', \text{argmax}_{a_i'} Q_{\theta_1}(s_i', a_i',))$
        Update policy networks' parameters $\theta_1$, $\theta_2$ by minimizing the loss
          $(j \in \{1,2\})$:
        $\theta_j \leftarrow \theta_j - \alpha \frac{dL(\theta_j)}{d\theta_j}$,
        where $L(\theta_j) = \frac{1}{|N|} \sum_i \left( Q_{\theta_j}(s_i, a_i) - y_i \right)^2$
        Update target networks' parameters $\theta_1^-$, $\theta_2^-$ by using soft update
          $(j \in \{1,2\})$:
        $\theta_j^- \leftarrow (1-\tau)\theta_j^- + \tau\theta_j$
    end
end
```

3.2. Testing Results of Different Q-learning-based RL

In order to test the robustness of different RL methods, we deliberately increase the challenge in looking for the real optimal solution through adding an artificial Gaussian noise δ to the original Q function at each training step before doing the arg max operation during the learning process:

$$Q_\theta^{\text{noisy}}(s', a_k) = Q_\theta(s', a_k) + \delta_k \tag{5}$$

where the additive noise is sampled from a normal distribution, i.e., $\delta_k \sim \mathcal{N}(0, \sigma_k)$. The standard deviation of σ_k is defined as:

$$\sigma_k = \beta \sqrt{\frac{1}{n} \sum_{i=k}^{n} \left(Q_\theta(s', a_k) - \bar{Q}_\theta(s') \right)^2} \tag{6}$$

where n is the size of the discrete action space, $Q_\theta(s', a_k)$ is the approximated Q value for an specific action a_k, $\bar{Q}_\theta(s')$ is the mean of Q values of all discrete actions at one specific state s' and β is a factor. After adding the noise, the ε greedy action a^* is selected based on Q_θ^{noisy} during the training process, i.e., $a^* = \arg\max_{a'}(Q_\theta^{\text{noisy}}(s',a'))$ with probability of $1-\varepsilon$.

To visualize the benefit of the clipped version of double DQN with added training noise, we compared its performance with other Q-learning methods by testing them for the traffic light management problem, which is shown in Fig. 3.

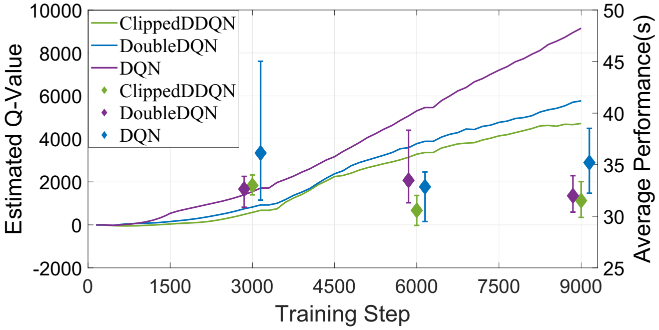

Figure 3. Comparison of the approximated Q values and observed average performance by using different Q-learning-based algorithms for TLM. The diamond markers displays the average performance of TLM, which is evaluated by the average waiting time of all vehicles at the intersection in five simulation tests and each lasts for one hour in simulation. The traffic density is set to 2200 vehicle/hour for these tests.

Figure 3 illustrates the overestimation issue of Q-learning, where the estimated Q value of the standard DQN increases steadily from the training step 3000 to 6000, while its actual performanc6ormance of DQN, results of three simulation tests with different random seeds, in which DQN is applied after it has been trained for 3000 steps, are shown in Fig. 4. The reward trajectories plotted in this figure are the observed values according to (1). It is noticeable that the observed rewards have large negative values in two tests as the vehicles suffered from longer waiting time in these tests, which again explains the poorer performance of DQN shown in Fig 3.

Based on the performance comparison, the clipped DDQN is selected and benchmarked with other traditional TLM methods in the next section.

4. TESTING RESULTS OF INTELLIGENT TRAFFIC LIGHT MANAGEMENT

4.1. Brief Introduction of The Baseline

Two types of TLM are used as the baseline for testing the performance of RL-based TLM. One is the fixed-time TLM. Its working principle is illustrated in Fig. 5. With the fixed-time TLM, each phase is selected sequentially and the green light can be applied to two phases at the same time. The only adjustable parameter is the duration of the green light for each phase. It needs to noted that the traffic light must be red for 6 sec before the green light is switched so that the vehicles from one phase have enough time to pass the intersection.

The other type of TLM used as the baseline is adaptive rule-based TLM. The detailed explanations on this TLM is given in our previous work [23]. The main idea of this

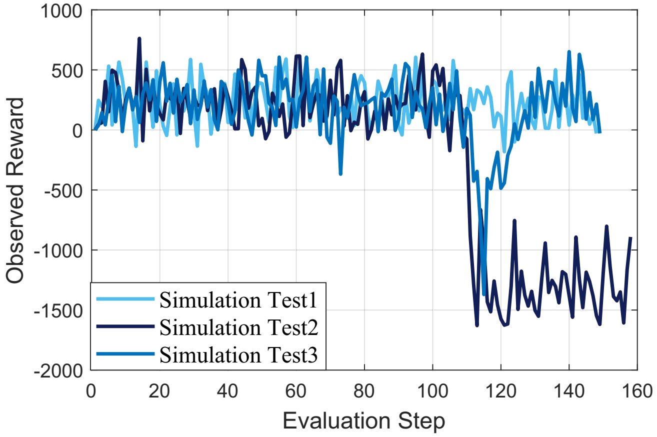

Figure 4. Observed reward trajectories of the DQN-based TLM after it has been trained by 3000 steps in three random simulation tests. The length of each simulation test is one hour.

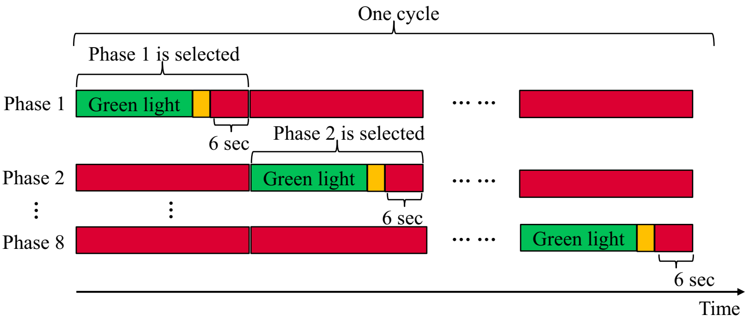

Figure 5. Working principle of the fixed-time TLM

method is to assign the green light to the phase which has the largest priority value, which is calculated by taking different traffic information (number of vehicles, waiting time of the cars in front of the intersection, vehicle speed, etc.) into account, as shown in the equation below,

$$\begin{aligned} Priority = & w_1 \times \frac{N_{\mathrm{veh},j}}{N_{\max}} + w_2 \times \frac{N_{\mathrm{precede},j}}{N_{\mathrm{maxQueue}}} + w_3 \times \frac{t_{\mathrm{wait},j}}{t_{\mathrm{maxWait}}} \\ & + w_4 \times \frac{v_j}{v_{\max}} - w_5 \times \frac{t_{\mathrm{delay},j}}{t_{\mathrm{maxDelay}}} \end{aligned} \tag{7}$$

where w_* is the weight for each item, j here is the index of vehicles driving in the same phase. Both selection of phases and the duration of the green lights are adjustable with this adaptive rule-based TLM. A simplified flowchart is shown in Fig. 6. The calibratable threshold min_{green} is the minimum time interval of being green. For instance, when this value is 10 sec, it means that the phase is not allowed to be changed before the current phase has lasted for at least 10 sec.

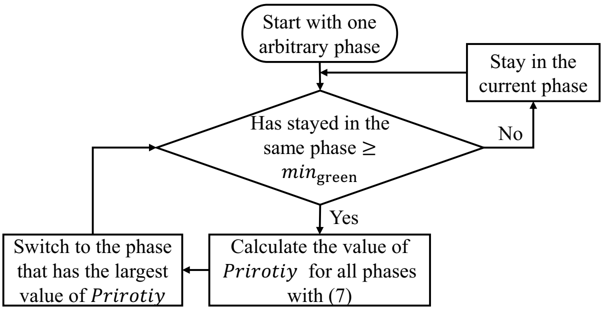
Start with one arbitrary phase
Stay in the current phase
Has stayed in the same phase ≥ min_{green}
No
Yes
Switch to the phase that has the largest value of *Prirotiy*
Calculate the value of *Prirotiy* for all phases with (7)

Figure 6. Simplified flowchart of the adaptive rule-based TLM

4.2. Performance Analysis of Different TLM Approaches

Figure 7 demonstrates the progress in training the clipped version of DDQN for intelligent TLM. It can be seen that the average reward of the last 100 steps, which is plotted with the red line in the figure, increases gradually and converges to a quasi-steady point after around 2000 training steps.

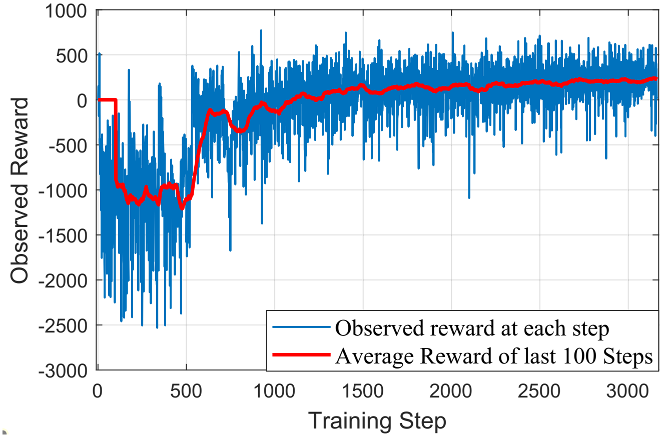

Figure 7. Training process of clipped version of DDQN for the intelligent TLM.

The performance of RL-based TLM is compared with fixed-time TLM, as demonstrated in Fig. 8. To check the performance of TLM in different traffic conditions, the traffic simulation is made with 5 traffic densities in Vissim, ranging from 1000 to 2600 vehicles per hour. It is noticeable that the fixed-time TLM with shortest time interval for the green light, has very poor performance when vehicle density is higher than 2200 vehicles per hour, behind which the main reasons are:

a) setting a shorter time interval for green traffic lights means more frequent switches between different phases, which results in more frequent deceleration and acceleration of the vehicles at the intersection and reduced traffic efficiency;
b) between the switches of the phases, the traffic lights are set to red for 6 sec for all driving directions for safety reasons. Therefore, the more switches means the

greater ratio between the total time interval of red lights and the total simulation time.

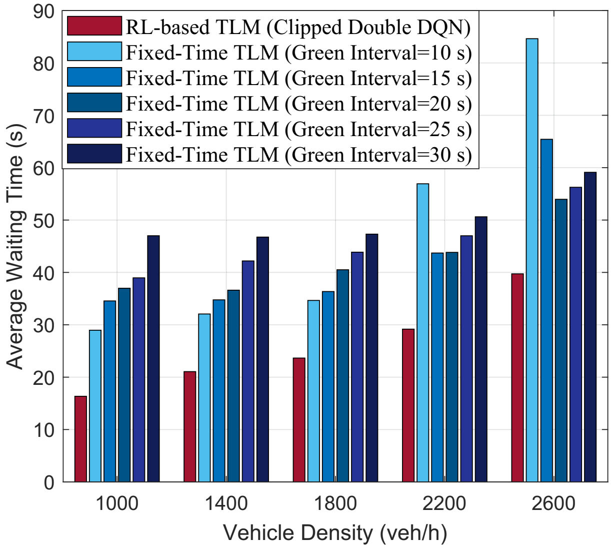

Figure 8. Performance comparison in terms of average waiting time of all vehicles within one hour simulation time between RL-based TLM and fixed-time TLM with different settings of the green light's interval.

In contrast, when the green traffic light interval is set to a bigger value, the performance of the fixed-time TLM is better when the traffic is dense, while its performance is unsatisfying when the traffic density is lower. This is because when the green light lasts unnecessarily for too long time for one phase, the vehicles from other driving directions which have a conflict with this phase have to stop for waiting.

In addition to being compared with fixed-time TLM, RL-based TLM is benchmarked with adaptive rule-based TLM as shown in Fig. 9.

With the adaptive rule-based TLM, intuitively, we expect to observe higher performance when a smaller threshold min_{green} in Fig. 6 is applied, because all possible actions, which can be taken with an arbitrary threshold, are also allowed to be taken when this threshold is smaller, but not vice versa. Therefore, Fig. 9 shows the trend that the average waiting time decreases with smaller min_{green} when the traffic density is below 2600 vehicles per hour. However, the performance of this adaptive rule-based TLM with the smallest threshold (i.e., when min_{green}=10 sec) suddenly deteriorates when the traffic density reaches 2600 vehicles per hour. This behavior seems to be counter-intuitive. The root reason behind this phenomenon is that the adaptive rule-based TLM behaves in a greedy way, i.e., the green light is set to the phase that has the largest priority value calculated with (7). Since (7) represents only the instantaneous status of each phase and it does not include the impact of current action on the traffic's future status into account, with a smaller value of the threshold min_{green}, this adaptive TLM is allowed to switch the phases more frequently in a greedy way. For instance, instead of keeping the traffic light being green for a longer time in one phase, the green light is switched to another phase whose priority value is slightly higher. This switch seems reasonable for one control

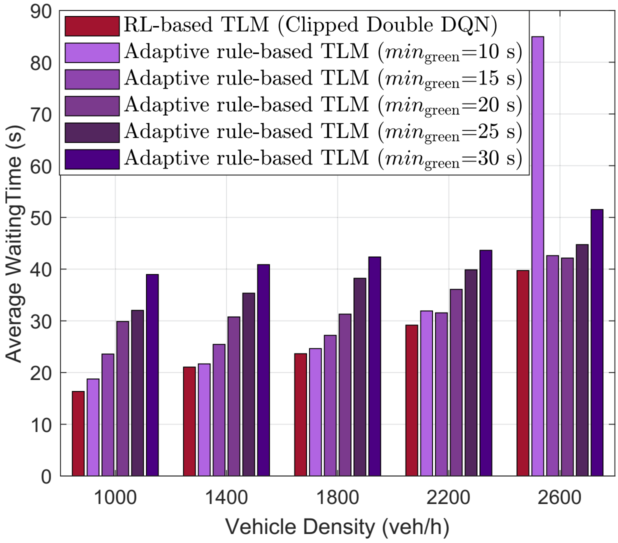

Figure 9. Performance comparison in terms of average waiting time of all vehicles within one hour simulation time between RL-based TLM and adaptive rule-based TLM with different settings of the minimum green light's interval.

step, but in fact it can be harmful for the traffic efficiency in the long term. In contrast, RL-based TLM has an intrinsic ability to make an optimal decision by considering not only the immediate reward at the current step, but also the expected total rewards in future. Therefore, in Fig. 9, we can see that RL-based TLM performs better than rule-based TLM in all different vehicle density conditions.

5. Conclusion

To sum up, a clipped version of double DQN is applied in this work for reducing the overestimation issue of Q-learning based reinforcement learning (RL) and its advantage over other Q-learning methods is demonstrated. The new RL algorithm enables the intelligent traffic light management (TLM) to achieve higher performance in terms of traffic efficiency, compared to both fixed-time and adaptive rule-based TLM.

Regarding the future work, we are considering two research directions. One direction is to get a similar control performance but with reduced interactions between the agent and environment, when the network that has been trained at one intersection is then used in a similar but new intersection. The other direction is the development of multi-agent RL for extending the application target from an isolated intersection to multi-intersection traffic signal control tasks.

References

[1] P. Koonce and L. Rodegerdts, "Traffic signal timing manual," United States. Federal Highway Administration, Tech. Rep., 2008.

[2] B. S. Kerner, *Introduction to modern traffic flow theory and control: the long road to three-phase traffic theory*. Springer Science & Business Media, 2009.

[3] B.-L. Ye, H. Gao, L. Li, K. Ruan, W. Wu, and T. Chen, "A MILP-based MPC method for traffic signal control of urban road networks," in *2019 Chinese Automation Congress (CAC)*. IEEE, 2019, pp. 3820–3825.

[4] C. Cav, A. Colak, and N. F. Unver, "Adaptive traffic signal control to reduce delay time at a single intersection point," in *2021 International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*. IEEE, 2021, pp. 1–6.

[5] M. Guo, P. Wang, C.-Y. Chan, and S. Askary, "A reinforcement learning approach for intelligent traffic signal control at urban intersections," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 4242–4247.

[6] M. E. M. Ali, A. Durdu, S. A. Çeltek, and A. Yilmaz, "An adaptive method for traffic signal control based on fuzzy logic with webster and modified webster formula using SUMO traffic simulator," *IEEE Access*, vol. 9, pp. 102 985–102 997, 2021.

[7] H. Nakanishi and T. Namerikawa, "Optimal traffic signal control for alleviation of congestion based on traffic density prediction by model predictive control," in *2016 55th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*. IEEE, 2016, pp. 1273–1278.

[8] J. Guo and I. Harmati, "Optimization of traffic signal control with different game theoretical strategies," in *2019 23rd International Conference on System Theory, Control and Computing (ICSTCC)*. IEEE, 2019, pp. 750–755.

[9] J. Luo, X. Li, and Y. Zheng, "Researches on intelligent traffic signal control based on deep reinforcement learning," in *2020 16th International Conference on Mobility, Sensing and Networking (MSN)*. IEEE, 2020, pp. 729–734.

[10] J. Zeng, J. Hu, and Y. Zhang, "Training reinforcement learning agent for traffic signal control under different traffic conditions," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 4248–4254.

[11] V. H. Pham and H.-S. Ahn, "Distributed stochastic model predictive control for an urban traffic network," *arXiv preprint arXiv:2201.07949*, 2022.

[12] S. Li, C. Wei, X. Yan, L. Ma, D. Chen, and Y. Wang, "A deep adaptive traffic signal controller with long-term planning horizon and spatial-temporal state definition under dynamic traffic fluctuations," *IEEE Access*, vol. 8, pp. 37 087–37 104, 2020.

[13] N. Wu, D. Li, Y. Xi, and B. De Schutter, "Distributed event-triggered model predictive control for urban traffic lights," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 4975–4985, 2020.

[14] H. Pang and W. Gao, "Deep deterministic policy gradient for traffic signal control of single intersection," in *2019 Chinese Control And Decision Conference (CCDC)*. IEEE, 2019, pp. 5861–5866.

[15] J. Zeng, J. Hu, and Y. Zhang, "Adaptive traffic signal control with deep recurrent Q-learning," in *2018 IEEE intelligent vehicles symposium (IV)*. IEEE, 2018, pp. 1215–1220.

[16] H. Yang, H. Zhao, Y. Wang, G. Liu, and D. Wang, "Deep reinforcement learning based strategy for optimizing phase splits in traffic signal control," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2022, pp. 2329–2334.

[17] F. Mao, Z. Li, Y. Lin, and L. Li, "Mastering arterial traffic signal control with multi-agent attention-based soft actor-critic model," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 3, pp. 3129–3144, 2022.

[18] C. Szepesvári, "The asymptotic convergence-rate of q-learning," *Advances in neural information processing systems*, vol. 10, 1997.

[19] H. Hasselt, "Double Q-learning," *Advances in neural information processing systems*, vol. 23, 2010.

[20] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, no. 1, 2016.

[21] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *International conference on machine learning*. PMLR, 2018, pp. 1587–1596.

[22] J. E. Smith and R. L. Winkler, "The optimizer's curse: Skepticism and postdecision surprise in decision analysis," *Management Science*, vol. 52, no. 3, pp. 311–322, 2006.

[23] Y. Jia, T. Al-Nusairi, and D. Görges, "Optimized traffic lights management algorithm with multiple information inputs for higher traffic and energy efficiency," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2022, pp. 2614–2619.