DERLight: A Deep Reinforcement Learning Traffic Light Control Algorithm with Dual Experience Replay

Zhichao Yang[1], Yan Kong[1], Chih-Hsien Hsia[2,3]*

[1] School of Computer and Software, Nanjing University of Information Science and Technology, China
[2] Department of Computer Science and Information Engineering, National Ilan University, Taiwan
[3] Department of Business Administration, Chaoyang University of Technology, Taiwan
yangzhichao319@163.com, kongyan4282@163.com, hsiach@niu.edu.tw

Abstract

In recent years, with the increasingly severe traffic environment, most cities are facing various traffic congestion problems, and the demand for intelligent regulation of traffic signals is also increasing. In this study, we propose a new intelligent traffic light control algorithm, dual experience replay light (DERLight), which innovatively and efficiently designs a dual experience replay training mechanism based on the classic deep Q network (DQN) framework and considers the dynamic epoch function. As results show that compared with some state-of-the-art algorithms, DERLight can shorten the average travel time of vehicles, increase the throughput at intersections, and also speed up the convergence of the network. In addition, the design of this algorithm framework is not only limited to the field of intelligent transportation, but also has transferability for some other fields.

Keywords: Deep reinforcement learning, Traffic light control, Dual experience replay, Dynamic epoch function

1 Introduction

Currently, in most countries and areas, traffic congestion has not been alleviated. Reasonable and efficient intelligent signal light timing can effectively solve the problem of urban traffic congestion. Therefore, the intelligent regulation of traffic lights has attracted more and more attention from researchers from all walks of life [1-5].

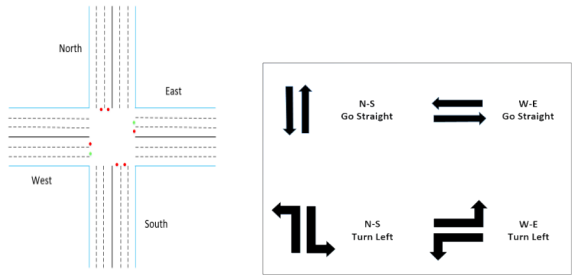
In the development history of signal lights, it has gone through three stages successively. The first is the traditional timing of signal lights, that is, there is only one set of fixed timing standards throughout the day. Then there is the dynamic signal light timing, that is, in different time periods, such as morning, noon, and evening, there are different timing standards. Compared with the traditional timing, although it has dynamic changes, it still cannot make scientific changes according to different traffic conditions. The last is the intelligent regulation of signal lights, that is, intelligent and scientific signal light timing based on the collected real-time traffic data. In these studies, algorithms based on deep learning [6-8] and reinforcement learning [9-11] have achieved certain results.

However, when the number of vehicles in the traffic network reaches a certain level, or when the scale of the road network is large, the effectiveness of some reinforcement learning (RL)-based algorithms begins to decrease, and the speed of training the network is also significantly slower. One of the reasons for this phenomenon is related to the RL-based algorithms themselves, and the other is that the design of network framework may affect the performance of algorithms. Some existing RL algorithms have the function of experience replay, which has indeed achieved great results in some cases [23-25]. However, when the conditions of the road network change rapidly, the training effect of the traditional experience replay will be greatly weakened. In this case, the concept of priority experience replay (PER) is introduced into the reinforcement learning framework [12], which trains the network by probabilistically selecting prioritized experience samples. This not only speeds up the convergence of the network, but also prevents the network from overfitting. However, PER [12] stores samples in the form of a binary tree, which has certain limitations. On the one hand, when facing a non-sparse reward environment with complex interactions, which can be considered as a complex sample space, the number of samples will be large and the shape of the binary tree will become complicated consequently. On the other hand, since PER [12] does not discards samples, it will waste some space resource of the replay buffer more or less.

Against the above, we used dual experience replay to train the network. To evaluate our dual experience replay idea, it is applied into traffic light control aiming at alleviating traffic congestion. Moreover, in order to make the network training more efficient, a dynamic epoch training mechanism is also introduced. The main contributions of this work include: 1) A traffic light control algorithm, dual experience replay light (DERLight) is proposed, which adopts the method of dual experience replay training to make up for the defect of PER [12]. Compared to some state-of-the-art algorithms, DERLight is better at reducing the average travel time of vehicles and increasing the average throughput at each intersection. 2) A dynamic epoch training mechanism is proposed, that is, the real time epoch value can be generated when the training is required, which is helpful for better network training. 3) The design of dual experience replay and dynamic epoch training mechanism proposed in

*Corresponding Author: Chih-Hsien Hsia; E-mail: hsiach@niu.edu.tw
DOI: 10.53106/160792642024012501007

this paper can theoretically be transferred to other fields, not limited to the field of intelligent transportation.



(a) The traffic intersection (b) 4 types of phases

Figure 1. An intersection

2 Related Work and Problem Definition

Traffic signal control problems strictly follow and could be modeled into Markov decision process problems (MDPs), and RL algorithms have become a research hotspot to solve this problem.

Among existed RL algorithms, experience replay-based ones are widely used, such as deep Q network (DQN), double DQN (DDQN), and deep deterministic policy gradient (DDPG), in which the design of the experience replay affects the efficiency of the entire network training to a large extent. On the basis of the traditional experience replay, some researches introduce the concept of shared experience replay into the RL models [13-15], so that each agent can share the global experience. While this enhances the connection between the agents, it is not so obvious in improving the efficiency of network training. For this case, [12] proposed prioritized PER. Compared with the traditional experience replay, it does have a more obvious advantage to accelerate the network convergence in many cases. However, PER [12] is not qualified enough in complex sample spaces, besides the reasons which have been introduced in the introduction section, another possible reason is that PER [12] must update the priority values of samples before selecting them. When the replay buffer is large, it causes bad effects on the time efficiency of network training, and this is unsatisfactory especially for time sensitive applications, e.g., the intelligent signal light control.

This work aims to study the intelligent regulation of signal lights in urban traffic networks. A transportation network contains more than one intersection, and the schematic diagram of an intersection is shown in Figure 1(a). An intersection has four directions ("W", "E", "N", "S"), each of which has 6 lanes, including incoming and outgoing lanes. The red and green dots in Figure 1(a) represent the red and green signal lights respectively. The phases of the traffic lights are combined in pairs and are divided into four groups, as shown in Figure 1(b).

Incoming and outgoing lanes. We set two kinds of lanes for the intersection, namely, the incoming lane and the outgoing lane, which are respectively defined as the roads where vehicles enter and leave the intersection. The incoming lanes into an intersection have three different directions: going straight, turning left, and turning right.

Traffic movement. Traffic movement reflects the trajectory of vehicles entering and leaving the intersection. If a car enters an intersection from lane a and exits from lane b, the traffic movement is recorded as (a, b). There are 3 lanes entering in one direction, thus four directions totally have 12 lanes, corresponding to 12 different types of traffic movements.

Signal phase. A signal phase is a traffic-directing measure taken at an intersection to allocate the right-of-way of traffic in time. At the intersection, there are four sets of phases, which are the straight phase and the left phase in the N-S direction and the E-W direction, respectively, as shown in Figure 1(b). The signal light for the right phase is always green.

Max pressure. In this study, we design the reward function with the concept of max pressure [16], which is defined as the difference value between the numbers of vehicles entering and leaving the lane respectively. The pressure can reflect the mutual influence of traffic flow between adjacent intersections, and is calculated as:

$$P_i = N_{in} - N_{out} \ . \tag{1}$$

where P_i denotes the max pressure of traffic movement i, N_{in} and N_{out} are the numbers of vehicles on the incoming and outgoing lanes respectively.

3 Proposed Methodology

3.1 DERLight Algorithm

In the DERLight framework, an agent is set up at each intersection of the traffic network to control the traffic signals at this intersection, and the control process is modeled into a MDP. The interaction between the agent and the environment is recorded in the form of a five-tuple, $<S, A, P, R, \gamma>$, where S represents the state space of the current interaction, A denotes the action space that the agent could take, P means the probability matrix of state transitions, R is the corresponding reward, and γ represents the discount factor.

The observation of an agent, including the number of vehicles and the condition of signal lights in each lane, is used as the current state of the intersection, which largely reflects the congestion situation of the intersection at that moment. According to the observation, the agent selects an action from the action space to adjust the corresponding signal phase, to alleviate the traffic congestion. When the agent selects an action, it will select the most suitable phase adjustment scheme according to the pressure value calculated by Eq. (1). Since there is a negative correlation between pressure and reward, pressure-based reward is defined as:

$$r_i = -P_i \ , \tag{2}$$

where P_i is the max pressure of traffic movement i, defined in Eq. (1). Therefore, the total reward for all the traffic movements due to the action a_t at state s_t is:

$$R(s_t, a_t) = \sum r_i \ . \tag{3}$$

DERLight adopts DQN as the framework for intelligent control of signal lights. We assume that the agent is currently at state s_t, and the Q value of taking action a_t at this moment is recorded as $Q(s_t, a_t)$, then:

$$Q(s_t, a_t) = R(s_t, a_t) + \gamma * \max\{Q(s_{t+1}, a_{t+1})\}, \quad (4)$$

where γ denotes the discount factor and ranging from 0 to 1 is the, representing the impact of the current action on the future. The more closer to 1, the more influence the current action has on the future. s_{t+1} represents the arrived state after taking a_t at state s_t, and a_{t+1} denotes the action taken at state s_{t+1}. The agent will choose the action with the highest Q value.

In this work, the loss function is used for gradient descent to update the parameters of the Q network and takes the form of mean square error as follows:

$$L = \sum \frac{1}{B}(R_t + \gamma \max Q'(s_{t+1}, a_{t+1}; \theta') - Q(s_t, a_t; \theta))^2, \quad (5)$$

where *Q' and Q* are the target network and evaluate network in DQN, respectively.

3.2 Dual Experience Replay

We design two experience pools. The basic idea of the first experience pool is consistent with the traditional one [17] in this work. The four tuples (s_t, a_t, r, s_{t+1}) obtained by the interaction between the agent and the environment are stored in the experience pool, and a batch is randomly taken from it when training is required, and the Q network is trained by Eq. (5).

The second experience pool is used to store priority experience, that is, interaction records with better effects, which are also stored in the form of four tuples. The judgment condition for priority experience is that its immediate reward must be greater than or equal to the previous average reward, and greater than the median value of the reward range. Compared with PER [12] using temporal difference (TD)-error to judge the priority experience, even if the judgment condition we designed is not as accurate as PER [12] that make up for the defect that PER [12] is not applicable in some cases.

When the first experience pool trains the network, the second one will have a probabilistic startup mechanism, the reason for this is to avoid the network going into overfitting. Before the start of the second experience pool, an epoch value is automatically generated, which is based on the reward at the current moment, the reward at the previous moment, and the average reward since the beginning of the round. It helps to train the network efficiently, while appropriately reducing unnecessary training time. Based on the idea of Taylor's formula [18], that is, any function can be approached in a polynomial, we fit r_t, r_{t-1}, and $r_{average}$ into a polynomial form, as follows:

$$epoch = epoch' + \lfloor \omega_1(r_t - r_{t-1}) \rfloor + \lfloor \omega_2 r_{average} \rfloor, \quad (6)$$

where *epoch* is the number of times that all samples need to be trained during the current training, and *epoch'* is the value of *epoch* at the last training before. ω_1 and ω_2 are two dynamic coefficients, which will change with r_t, r_{t-1}, and $r_{average}$. The initial value of *epoch* is set to 1000. The value of ω_1 and ω_2 should first be related to time. When the number of vehicles in the early stage is relatively small, that is, when the $r_{average}$ does not have much meaning, the value of ω_1 should be increased and the value of ω_2 should be decreased, and the opposite is true in the later stage. For this problem, we consider using the inverse tangent function. The reason is that it not only satisfies the correlation with time, but also guarantees that its value range is bounded. Meanwhile we consider that the total duration of a traffic dataset is 60 minutes, the inverse tangent function as shown in Figure 2. And we also consider that the value of ω_1 and ω_2 should be inversely proportional to (r_t-r_{t-1}) and $r_{average}$. To sum up, the formula we designed is as follows:

$$\omega_1 = -arctan(30 - T) * (r_{t-1} - r_t), \quad (7)$$

$$\omega_2 = -arctan(T - 30) * r_{average}, \quad (8)$$

where T represents the time, and its range is [0,60].

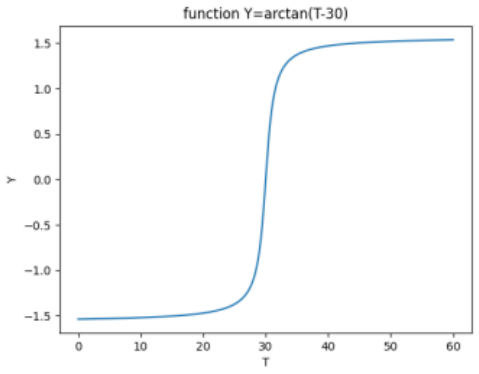


Figure 2. The inverse tangent function

The network framework of DERLight is shown in Figure 3, and its pseudocode is shown in Algorithm 1. The difference between it and DQN lies in the setting of the second experience replay, which uses the same loss function for training. At the same time, their asynchronous training greatly reduces the probability of the network falling into local optima.

In summary, the design of the dual experience replay is not only to improve the efficiency of sampling, but also to implement the replay function of priority experience in a new way. And the dynamic epoch mechanism reduces unnecessary training time to a certain extent.

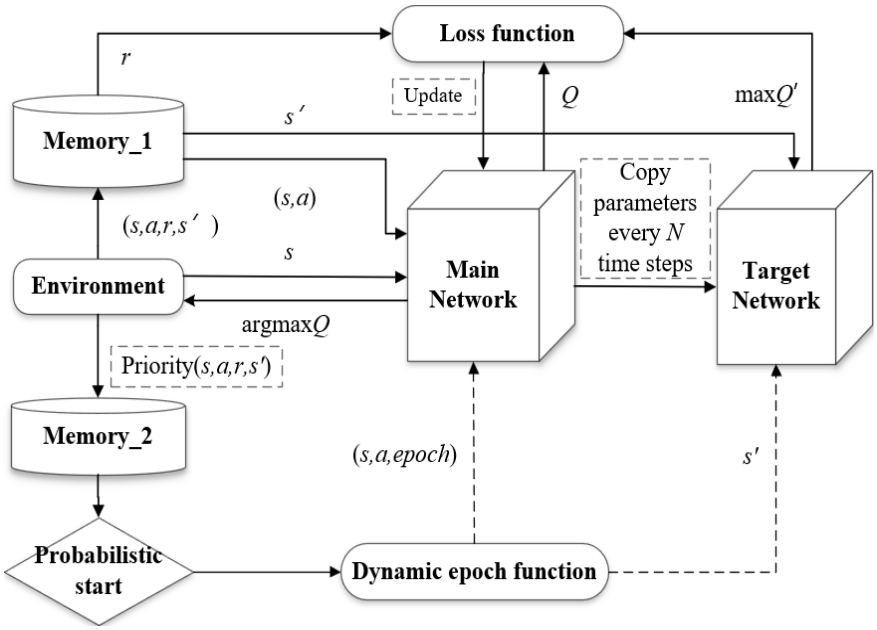


Figure 3. The network framework of DERLight

Algorithm 1. DERLight

Input: initial replay memory $\mathcal{D}$, new replay memory $\mathcal{D}$', sample size B, episode length T, discount factor γ, greedy ϵ, learning rate α, replacement frequency C, number of vehicles in the incoming and outgoing lanes N_{in} and N_{out}, maximum carrying capacity in the incoming and outgoing lane N_{maxin} and N_{maxout}

Initialize Q with parameters $\boldsymbol{\theta}$, Q' with parameters $\boldsymbol{\theta'}$
for each *episode* **do**
  Initialize step number t as 0, total time t_{sum} as 0;
  while $t_{sum} < T$ **do**
    Select a random phase h with probability ϵ ;
    Otherwise $h \leftarrow argmax_h Q(s_t, h; \theta)$;
    Observe the green phase duration time t_g from the environment;
    Execute $a_t \leftarrow \{h, t_g\}$;
    Observe the next state s_{t+1};
    Calculate the reward based on max pressure
    $\boldsymbol{R_t} \leftarrow -(\boldsymbol{N_{in}} - \boldsymbol{N_{out}})$;
    Store quadruple (s_t, a_t, R_t, s_{t+1}) in $\mathcal{D}$;
    Calculate the average and the median reward
    $\boldsymbol{R_{average}} = \sum_{i=0}^{t} \boldsymbol{R_i}$
    $\boldsymbol{R_{median}} \leftarrow (\boldsymbol{N_{maxout}} + \boldsymbol{N_{maxin}})/2$;
    if $R_t >= R_{average}$ && $R_t > R_{median}$ **then**
      Store quadruple (s_t, a_t, R_t, s_{t+1}) in $\mathcal{D}$';
    $t_{sum} \leftarrow t_{sum} + t_g$, $t \leftarrow t + 1$.
    if $|h| > B$ **then**
      Select B samples from $\mathcal{D}$ randomly;
    end if
    Calculate the loss L by Eq. (5) and update θ by Gradient Descent with learning rate α;
    if *rand* (0,9) > 1 **then**
      Select B samples from $\mathcal{D}$' randomly;
      Calculate the loss L by Eq. (5) and update θ by Gradient Descent with learning rate α;
    end if
    Every C steps update $Q' \leftarrow Q$.
  end while
end for

4 Experimental Results

4.1 Datasets

In this section, DERLight is evaluated in a widely used simulation platform named CityFlow [19]. DERLight is evaluated totally on eight datasets, including both synthetic and real-world ones, as shown in Table 1.

Table 1. Traffic flow data

Traffic Flow	Interval	Volume
Flow-Light-1	1×6	4460
Flow-Light-2	1×6	4887
Flow-Heavy	1×6	8895
Flow-Jinan	3×4	6281
Flow-NewYork-1	1×16	6689
Flow-NewYork-2	1×16	3955
Flow-NewYork-3	1×16	5992
Flow-NewYork-4	1×16	4405

To test DERLight on the synthetic data, we choose a 1×6 traffic network, *i.e.*, 1 road in the E-W direction and 6 roads in the N-S direction. In specific, this work uses three different traffic flows (*i.e.*, Flow-Light-1, Flow-Light-2, and Flow-Heavy in Table 1) to test the performance.

For the real-world datasets (*i.e.*, Flow-Jinan, Flow-NewYork-1, Flow-NewYork-2, Flow-NewYork-3, Flow-NewYork-4 in Table 1), a 16×1 and a 3×4 traffic networks are used. Real-world traffic flow data is more random than synthetic data, and the comparison is shown in Table 1, where interval indicates the scale of the current road network, and volume represents the number of traffic flow.

4.2 Experiment Settings and Benchmarks

For all intersections, we set the same signal timing. In the intelligent control of signal lights, the agent can only adjust the phase for 10 seconds each time. In the experiment, we set

3600 seconds as a round, the value range of ranges from 0.8 to 0.2, the discount factor is set to to be 0.8, the learning rate of the Q network is 0.001, and the target network is updated every 5 steps in this work. In addition, we set the maximum capacity of two experience pools to 10000. However, the second one does not have the function of discarding samples. All the above parameters have been explained in the pseudo of Algorithm 1.

PressLight [20]. A pressure-based RL algorithm. The reward function is designed with the concept of pressure, so as to achieve the purpose of mutual influence and cooperation between adjacent intersections.

CoLight [21]. An RL algorithm based on the Graph Attention Network. CoLight combines the Graph Neural Network and the attention mechanism and adopts the Graph Attention Network to reflect the connections between multiple intersections.

PDLight [22]. A pressure-based RL algorithm. On the basis of the original pressure concept, PDLight also considers the carrying capacity on the outgoing lane and proposed a new pressure calculation formula as the reward function.

4.3 Evaluation Metrics

To evaluate the performance of DERLight, average travel time, average throughput and average network training time are used as metrics.

Average Travel Time. In the experiment, we recorded the time of a vehicle entering and leaving the intersection, denoted as t_{enter} and t_{leave} respectively. Then the travel time of the vehicle at the intersection is $t_{leave} - t_{enter}$, so the average travel time for an intersection is the average travel time of all vehicles.

Average Throughput. Average throughput is defined as the average traffic volume at all intersections on the road network.

Average Network Training Time. We take the average training time of all rounds as an experimental metric to evaluate the time efficiency of the network training.

Table 2 and Table 3 show the average travel time and average throughput in different traffic networks, from which it can be seen that our proposed DERLight is optimal in almost all datasets, especially in real-world datasets. The average travel time of each traffic network can be clearly seen, and the performance of DERLight can also be seen from it from Table 2. In the 1 × 6 traffic network, both DERLight and CoLight can achieve lower average travel time under several traffic flow (*i.e.*, Light-1, Light-2 and Heavy in Table 2). However, as the scale of the traffic network increases, the performance of CoLight begins to decline. For example, in the 3 × 4 traffic network, the overall performance of CoLight is not as good as that of DERLight. When facing the larger traffic network, such as 1 × 16, the performance of CoLight and the other two algorithms are more unstable, and their average travel time is generally higher than that of DERLight. Table 3 shows the average throughput of each road network under different conditions. In the 1 × 6 traffic network, when faced with low traffic flow, such as Light-1 and Light-2, the performance differences of several algorithms are not significant, and DERLight can achieve a weak advantage. When faced with high traffic flow, the advantages of DERLight are more obvious (*i.e.*, Heavy in Table 3). From other traffic networks, it can also be seen that DERLight has certain advantages compared to other algorithms.

In order to compare the network training time, we recorded the data while the network was training, as shown in Table 4. Meanwhile we also recorded the change process of dynamic epoch in DERLight, as shown in Figure 4. From Figure 4, the fluctuation in the early stage of the curve is relatively large, but it usually stabilizes at a smaller value in the later stage. From the results, it can be seen that DERLight not only achieves better performance, but also reduces the average training time of the network as a whole.

Table 2. Average travel time (seconds)

	Light-1	Light-2	Heavy	Jinan	NewYork-1	NewYork-2	NewYork-3	NewYork-4
PressLight	33.89	36.51	38.66	72.09	20.60	19.26	22.24	20.40
PDLight	31.11	31.09	28.55	59.67	25.58	22.51	19.59	20.09
CoLight	27.08	**26.58**	28.17	55.01	22.94	20.82	20.12	20.02
DERLight	**26.87**	27.12	**28.10**	**46.85**	**18.27**	**12.12**	**12.71**	**18.68**

Table 3. Average throughput

	Light-1	Light-2	Heavy	Jinan	NewYork-1	NewYork-2	NewYork-3	NewYork-4
PressLight	1257	1275	2602	1169	912	866	728	1002
PDLight	1258	1277	2570	**1359**	844	846	624	967
CoLight	1255	1269	2487	1191	917	**943**	747	1024
DERLight	**1277**	**1282**	**2644**	1354	**921**	908	**777**	1045

Table 4. Average network training time (seconds)

	Light-1	Light-2	Heavy	Jinan	NewYork-1	NewYork-2	NewYork-3	NewYork-4
PressLight	36.05	36.22	46.06	59.79	92.61	90.28	90.73	91.98
PDLight	37.31	39.41	47.61	56.88	96.59	93.45	90.80	94.39
CoLight	38.33	38.43	48.13	58.61	**88.15**	92.91	89.96	90.73
DERLight	**35.64**	**35.63**	**41.97**	**54.50**	90.79	**86.99**	**89.60**	**88.86**

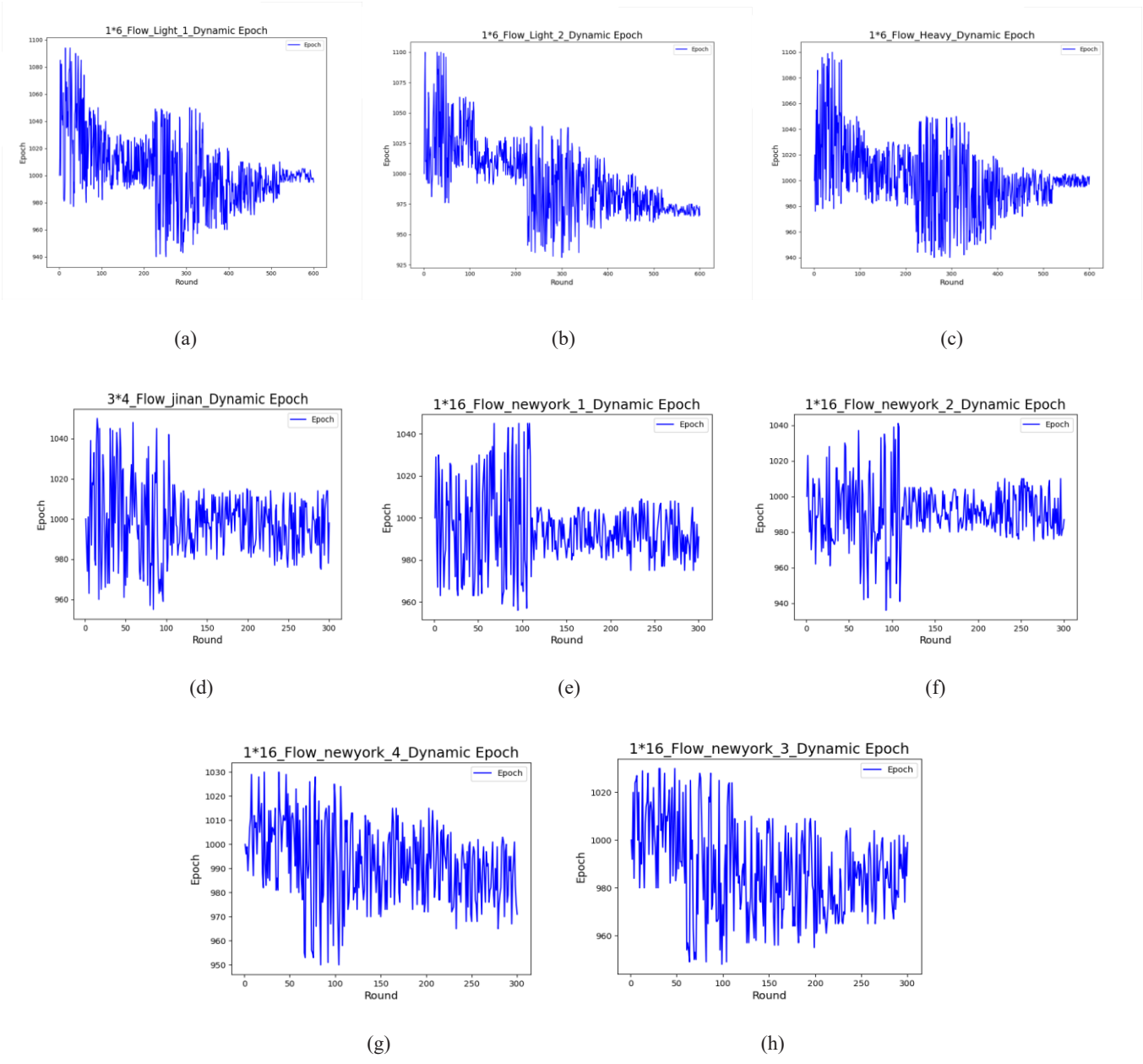


Figure 4. Dynamic epoch in DERLight under various traffic flows, which included the changes of epoch within 300 rounds

5 Conclusion

In this work, we propose a novel traffic light control algorithm, DERLight. Its innovation is mainly in the introduction of dual experience replay and dynamic epoch training mode. The results on the synthetic and real-world datasets show that DERLight can not only shorten the average travel time of vehicles, increase the average throughput at intersections, but also shorten the network training time. The detailed analysis of performance also demonstrates that DERLight's framework can theoretically be transferred to research in other fields, which means DERLight has both practical and theoretical significance. This paper provided some guidance for the future development of artificial intelligence, such as the innovation and efficiency of sampling, as well as the reinforcement learning model trained by dual experience replay.

References

[1] H. Wei, G. Zheng, H. Yao, Z. Li, Intellilight: a reinforcement learning approach for intelligent traffic light control, *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, London, United Kingdom, 2018, pp. 2496-2505, 2018.

[2] L. Kuyer, S. Whiteson, B. Bakker, N. Vlassis, Multiagent reinforcement learning for urban traffic control using coordination graphs, *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Antwerp, Belgium, 2008, pp. 656-671.

[3] C. Chen, H. Wei, N. Xu, G. Zheng, M. Yang, Y. Xiong, K. Xu, Z. Li, Toward a thousand lights: decentralized deep reinforcement learning for large-scale traffic signal control, *AAAI Conference on Artificial Intelligence*, Vol. 34, No. 4, pp. 3414-3421, April, 2020.

[4] X. Zang, H. Yao, G. Zheng, N. Xu, K. Xu, Z. Li, MetaLight: value-based meta-reinforcement learning for traffic signal control, *AAAI Conference on Artificial Intelligence*, Vol. 34, No. 1, pp. 1152-1160, April, 2020.

[5] W. Genders, S. Razavi, Policy analysis of adaptive traffic signal control using reinforcement learning, *Journal of Computing in Civil Engineering*, Vol. 34, No. 1, Article No. 0000859, January, 2020.

[6] B. Yu, H. Yin, Z. Zhu, Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting, *International Joint Conference on Artificial Intelligence*, Stockholm Sweden, 2018, pp. 3634-3640.

[7] G. Zheng, Y. Xiong, X. Zang, J. Feng, H. Wei, H. Zhang, Y. Li, K. Xu, Z. Li, Learning phase competition for traffic signal control, *ACM on Conference on Information and Knowledge Management*, Beijing, China, 2019, pp. 1963-1972.

[8] J. A. Laval, H. Zhou, Large-scale traffic signal control using machine learning: some traffic flow considerations, August, 2019. https://arxiv.org/abs/1908.02673

[9] I. Arel, C. Liu, T. Urbanik, A. G. Kohls, Reinforcement learning-based multi-agent system for network traffic signal control, *IET Intelligent Transport Systems*, Vol. 4, No. 2. pp. 128-135, June, 2010.

[10] S. El-Tantawy, B. Abdulhai, H. Abdelgawad, Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (MARLIN-ATSC): methodology and large-scale application on downtown Toronto, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 14, No. 3, pp. 1140-1150, September, 2013.

[11] X. Hu, C. Zhao, G. Wang, A traffic light dynamic control algorithm with deep reinforcement learning based on GNN prediction, September, 2020. https://arxiv.org/abs/2009.14627

[12] T. Schaul, J. Quan, I. Antonoglou, D. Silver, Prioritized experience replay, February, 2016. https://arxiv.org/abs/1511.05952

[13] D. Horgan, J. Quan, D. Budden, G. Barth-Maron, M. Hessel, H. v. Hasselt, D. Silver, Distributed prioritized experience replay, March, 2018. https://arxiv.org/abs/1803.00933

[14] S. Schmitt, M. Hessel, K. Simonyan, Off-policy actor-critic with shared experience replay, *International Conference on Machine Learning*, Vol. 119, pp. 8545-8554, 2020.

[15] C. Li, Y. Li, Y. Zhao, P. Peng, X. Geng, Sler: self-generated long-term experience replay for continual reinforcement learning, *Applied Intelligence*, Vol. 51, No. 1, pp. 185-201, January, 2021.

[16] P. Varaiya, the max-pressure controller for arbitrary networks of signalized intersections, in: S. Ukkusuri, K. Ozbay (Eds.), *Advances in Dynamic Network Modeling in Complex Transportation Systems- Complex Networks and Dynamic Systems*, vol 2, Springer, New York, NY, 2013, pp. 27-66.

[17] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller, Playing atari with deep reinforcement learning, December, 2013. https://arxiv.org/abs/1312.5602

[18] Z. M. Odibat, N. T. Shawagfeh, Generalized Taylor's formula, *Applied Mathematics and Computation*, Vol. 186, No. 1, pp. 286-293, March, 2007.

[19] H. Zhang, S. Feng, C. Liu, Y. Ding, Y. Zhu, Z. Zhou, W. Zhang, Y. Yu, H. Jin, Z. Li, CityFlow: a multi-agent reinforcement learning environment for large scale city traffic scenario, *ArXiv preprint*, *arXiv: 1905.05217*, May, 2019. https://arxiv.org/abs/1905.05217

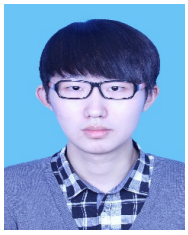
[20] H. Wei, C. Chen, G. Zheng, K. Wu, V. Gayah, K. Xu, Z. Li, PressLight: learning max pressure control to coordinate traffic signals in arterial network, *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Anchorage, AK, USA, 2019, pp. 1290-1298.

[21] H. Wei, N. Xu, H. Zhang, G. Zheng, X. Zang, C. Chen, W. Zhang, Y. Zhu, K. Xu, Z. Li, Colight: learning network-level cooperation for traffic signal control, *ACM on Conference on Information and Knowledge Management*, Beijing China, 2019, pp. 1913-1922.

[22] C. Zhao, X. Hu, G. Wang, PDLight: a deep reinforcement learning traffic light control algorithm

with pressure and dynamic light duration, September, 2020. https://arxiv.org/abs/2009.13711
[23] X. Guo, J. Yang, Z. Gang, A. Yang, Research on network security situation awareness and dynamic game based on deep Q learning network, *Journal of Internet Technology*, Vol. 24, No. 2. pp. 549-563, March, 2023.
[24] J. Zhang, C. Zhang, W. Chien, Overview of deep reinforcement learning improvements and applications, *Journal of Internet Technology*, Vol. 22, No. 2. pp. 239-255, March, 2021.
[25] Y. Chen, J. You, Effective radio resource allocation for IoT random access by using reinforcement learning, *Journal of Internet Technology*, Vol. 23, No. 5. pp. 1069-1075, September, 2022.

Biographies



Zhichao Yang received his M.S. degree in Computer Technology from Nanjing University of Information Science and Technology, China. His research interests include Deep Reinforcement Learning and Multi-agent System. Moreover, his research focuses on the intelligent traffic signal light control.

Yan Kong received her Ph.D. degree in Computer Science from the University of Wollongong, Australia. Currently, she works as a faculty in Nanjing University of Information, Science and Technology, China. Her research interests include Deep learning, Multi-agent system, and Machine Learning. Her research focuses on the smart control on the traffic signal lights to alleviate the traffic congestion.

Chih-Hsien Hsia received the Ph.D. degree in Electrical and Computer Engineering from Tamkang University, and the second Ph.D. degree from National Cheng Kung University, Taiwan, respectively. He currently is a Full Professor and a Chairperson with the Department of Computer Science and Information Engineering, NIU. His research interests include DSP IC Design, AI in Multimedia, and Cognitive Learning.