

Proposal of a Signal Control Method Using Deep Reinforcement Learning with Pedestrian Traffic Flow

Akimasa Murata, Yuichi Sei^a, Yasuyuki Tahara^b and Akihiko Ohsuga^c

The University of Electro Communications, Tokyo, Japan


Keywords: Agents, Deep Reinforcement Learning, Traffic Control.


Abstract: In dealing with traffic control problems, there have been studies on learning signal change patterns and timing by using reinforcement learning for signals. In most of them, the focus is on improving the delay time of vehicles, and few of them assume the traffic situation including pedestrians. Therefore, the objective of this study is to provide traffic control to reduce traffic delays for both vehicles and pedestrians in an environment where pedestrian traffic volume varies greatly. Then, we will verify the accuracy with traffic signals considering the temporal changes of the environment. Results of verification, although vehicle wait times increased, a significant reduction in pedestrian wait times was observed.


1 INTRODUCTION

In the modern traffic environment, vehicles and pedestrians are mixed, and traffic control is achieved by using traffic signals appropriate to the environment. In response to this, research is being conducted to create traffic signals that can respond to real-time traffic changes by learning signal control policies using DQN (Deep Q-Network)(Mnih et al., 2013), one of the reinforcement learning methods. Most of those studies focus on vehicles, but in the real environment, we need to pay attention to exceptional factors such as pedestrians. Information on the size of pedestrian groups and their destinations is an important factor in understanding the traffic environment and understanding of human flow should lead to appropriate traffic control. Therefore, this study focuses on pedestrian control and aims to develop a traffic control system that can respond to changes in the environment. Then, by creating traffic signals with a network with LSTM (Long Short-Term memory) (Hochreiter and Schmidhuber, 1997), (Gers et al., 2000) added. This attempt to control for the large differences in speed between vehicles and pedestrians, as well as the differences in where they are moving on the road. In this way, we verify the feasibility and the system of control based on temporal changes in traffic vol-

ume. Traffic signals that are controlled using the current traffic environment are referred to as DTC (DQN Traffic Control signal), while traffic signals that are controlled using past traffic conditions are referred to as LTC (LSTM Traffic Control signal). Traffic signals were created using deep reinforcement learning, and their accuracy was evaluated using DTC for traffic signals that are controlled based on the current environment and LTC for traffic signals that are controlled using past information on the environment. As a result of a comparison with FTC (Fixed Traffic Control signal), which switches signals in sequence, DTC reduces waiting time per vehicle by about 74% and increase waiting time per pedestrian by about 196%, DTC reduced overall vehicle and pedestrian waiting time by approximately 12%, and it can be concluded that DTC is able to control the environment according to the conditions of the environment. LTC was not learning to control well because it increased vehicle and pedestrian wait times. However, even in environments with large numbers of pedestrians, the waiting time per person did not increase significantly. This paper is organized as follows. Section 2 introduces related research. Section 3 describes the proposed method. Section 4 describes the experiment, the evaluation and its discussion. Finally Section 5 summarizes the conclusions of this paper.

^a  <https://orcid.org/0000-0002-2552-6717>

^b  <https://orcid.org/0000-0002-1939-4455>

^c  <https://orcid.org/0000-0001-6717-7028>

2 RELATED WORKS

2.1 DQN

Q-learning is one of the reinforcement learning methods. It is a learning method that uses a table called a Q-table to determine the value of an action in each situation. Based on the Q-value on the Q-table, the value of an action in each state is determined, and the next action is selected accordingly. The Q value is updated for each action based on the value of the action in each state and the value of the subsequent actions. However, the creation of a Q-table for estimating Q-values becomes more complicated as the number of states and actions increases. DQN takes in a neural network in the estimation of this Q-value and is capable of stable and accurate estimation.

2.2 R2D2

DRQN (Deep Recurrent Q-Learning) (Hausknecht and Stone, 2017) is a learning method that combines LSTM, a model that enables learning of long-term dependencies on time series data, and DQN. However, this learning method is incompatible with experience replay of DQN, and there was a problem that the hidden state of LSTM was initialized. R2D2 (Recurrent Experience Replay in Distributed Reinforcement Learning) (Kapturowski et al., 2019) is a method that solves this problem. During the simulation, hidden state of LSTM is saved with the experience saved, such as the state of the environment and the agent's rewards. The network is then updated during training. By allowing time to pass through only time-series data without learning for a certain period, this method enables hidden state to be close to the state when experience is saved.

2.3 Traffic Control Studies

Most of the studies on traffic control using traffic signals trained by reinforcement learning have been conducted using traffic simulators. The Simulation of Urban Mobility (SUMO) (Behrisch et al., 2018) is the main simulator used. The method used in traffic control is to learn to use traffic signals as agents by using tensors obtained from the presence or absence of vehicles in the traffic network (Vidali, 2017), (Liang et al., 2018). By passing the obtained tensor representing the traffic condition to the neural network, an action is selected. The value of the action is determined according to the change of the state by the action, and the network is updated by using it. In addition, a cooperative system that updates Q-values

by transferring Q-values of adjacent traffic lights to a multi-agent system (Ge et al., 2019) and a study introducing LSTM have also been conducted (Choe et al., 2018).

3 METHOD

3.1 Overview

In this study, we consider signal control in an environment where vehicles and pedestrians are present on the road and their traffic volume varies greatly. SUMO is used to create an environment in which the traffic volume changes, from which the traffic condition of the environment is obtained. Deep reinforcement learning is performed based on the results. In the traffic simulation, the durations of green and yellow lights are fixed, and the total time is set to k . In this study, the value of k is 20 seconds on the simulator. Traffic light acquires traffic conditions (location information of vehicles and pedestrians), selects actions, and calculates rewards of reinforcement learning at each time k . In the following description, the value that increases with each time k is defined as step t , and the list of obtained values is shown in Table 1.

Table 1: Symbol list.

Symbol	Expression
s_t	State of the environment at step t
w_t	Waiting time between step $t-1$ and step t
r_t	Reward between step $t-1$ and step t
a_t	Selected action at step t
M	Experiential memory

3.2 State Representation

The traffic light acquires information on the location of vehicles and pedestrians in the environment every time k . Then, the road in the vicinity of the signal is divided into segments of a predetermined distance. The state of the environment is represented by creating a list according to the number of vehicles and pedestrians moving or stopped within the segmented area. In addition, the number of pedestrians present in the pedestrian crossing at the intersection is added to the list. This is to treat pedestrians in crosswalks as information to determine whether a vehicle can turn right or left. The environment state list s_t at step t is represented as $s_t = \{N_t^1, N_t^2, \dots, N_t^{104}\}$. This is a

list with 104 elements and the number of vehicles or pedestrians in the road division range j with N_j^j . The vehicle division range is set to the same size as that of the vehicle in the three-lane portion near the signal. In the two-lane portion, the range is gradually expanded to twice, three times, and four times the size of the vehicle. The pedestrian division range is about half that of a vehicle, and is more segmented than that of a vehicle (Figure 1). This is because the size of each pedestrian in the environment is smaller than that of a vehicle, and the purpose is to clarify the density of pedestrians within the segmented area.

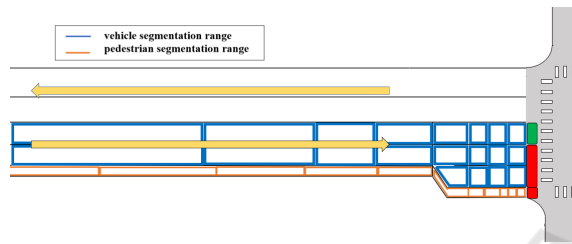


Figure 1: Segmentation range.

3.3 Action

The traffic light chooses its action using a state list obtained from the environment at each time k . The action is to maintain the current signal pattern or change to one of the signal patterns shown in Figure 2. In low traffic environments, simple control is possible with only P_0 and P_1 . However, in an environment with heavy pedestrian traffic, right and left turns by vehicles are restricted, so P_2 and P_3 are set to allow for this. In addition, P_4 , P_5 , and P_6 are provided for the case of extremely heavy vehicle or pedestrian traffic.

3.4 Reward

The reward values in reinforcement learning are mainly determined by the waiting time of vehicles and pedestrians near intersections. The waiting time of stopped vehicles and pedestrians is obtained for each step t . Assuming that the vehicle waiting time at step t is CW_t and the pedestrian waiting time is PW_t , the reward function R_t is expressed by the following equation 1

$$R_t = (CW_{t-1} - CW_t) + \alpha(PW_{t-1} - PW_t) - \beta EM_t \quad (1)$$

α is an arbitrary constant, and it's set up to adjust for the importance of pedestrians in the environment. In this study, α is set to 3 because the environment used in the study is an environment with many

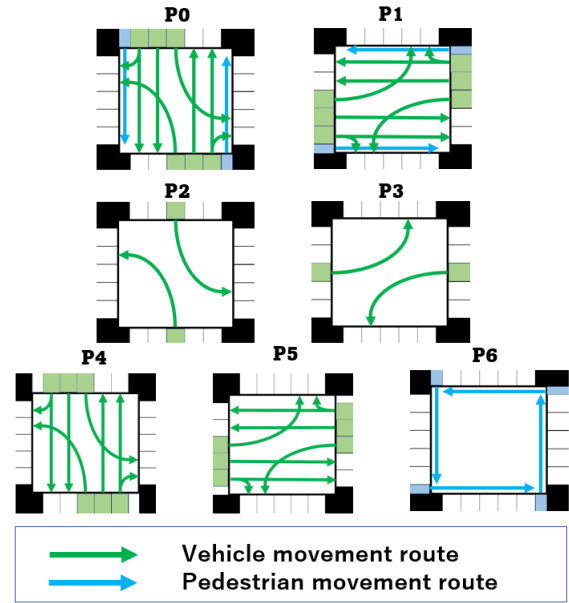


Figure 2: Signal pattern.

pedestrians. Also, EM_t in equation is the count of the number of emergency brakes the vehicle has caused from the previous step to the current step. Emergency braking may occur at the timing of a signal change or when pedestrians are present in the crosswalk. The emergency braking is considered to be the basis for causing accidents and was established with the aim of reducing the number of such accidents. β is a constant that is negative, and the reward decreases as the number of emergency brakes increases. In this study, β is set to -5000. This value was determined based on experiments with traffic signals that switch signals in sequence. The results of this experiment showed a very low value for the vehicle reward, so we set a very high value compared to that value. This was expected to promote the inhibition of emergency braking during learning.

3.5 Network

The network used in this study consists of five fully connected layers. In addition, in order to have traffic control using temporal information, we prepared a network including LSTM (Figure 3). This network is the one in which the first layer of the aforementioned network consisting only of fully connected layers is changed to the LSTM layer. In this network, the input is a tensor consisting of states over several steps, and depending on the state of the environment, one of the signal patterns shown in section 3.3 is obtained as an output.

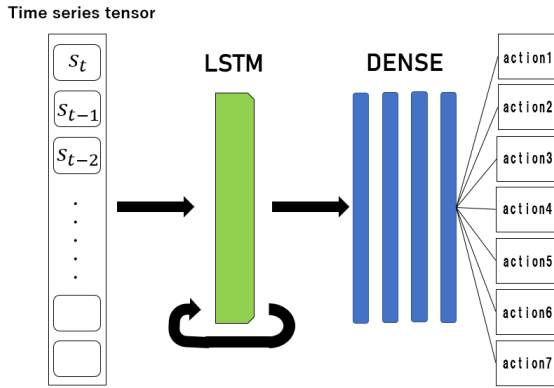


Figure 3: Overview of the network.

3.6 Algorithm

The learning algorithms Algorithm 1 and Algorithm 2 are as follows. Algorithm 1 shows the flow of learning by DQN and is based on the algorithm published by Andrea Vidali (Vidali, 2017). First, it runs the simulation with SUMO. Second, it obtains a state by number of vehicles and pedestrians in the environment. Third, using the obtained state list, the signal selects the action. Then, the algorithm calculates rewards based on vehicle and pedestrian waiting time. Fourth, save the acquired state and reward values to experience memory. Repeat this process for each step t . After a certain amount of time, the network is updated using the saved experience. Experimental memory stores the state of the environment and the value of rewards, the maximum amount of storage is defined. Therefore, if it is exceeded, the oldest data is deleted. The reason for determining the maximum amount of storage is that ϵ -greedy makes the older data highly random. Algorithm 2 shows learning including LSTM. In learning with this network, R2D2 (Kapturowski et al., 2019) methods were incorporated. Stores the hidden state of the LSTM layer at each step t in the experience and updates the network with it during learning. Also, do not use the time series tensor for a few steps for training, but just pass it through the network. This ensures that the hidden state of LSTM is close to the state when the time series tensor is saved.

4 EXPERIMENT

4.1 Test Environment

In this study, we performed simulations using SUMO and evaluated the accuracy of traffic signals. Simulation was performed up to 100 episodes, with 4000 steps on SUMO as one episode. The environment

Algorithm 1: Learning Algorithm.

```

1: for episode = 1 to  $N$  do
2:   while step < max_step do
3:     get traffic states and waiting_time:  $s_t, w_t$ 
4:     estimate reward :  $r_t$ 
5:     update old adjacent signal action
6:     if len( $M$ ) > max_size - 1 then
7:       delete  $M[0]$ 
8:     end if
9:     append experience = ( $s_t, a_{t-1}, r_t, s_{t-1}$ )
10:    select action with  $\epsilon$ -greedy :  $a_t$ 
11:    yellow phase and green phase
12:     $t = t + 1$ 
13:    update  $s_{t-1} = s_t, a_{t-1} = a_t$ 
14:  end while
15:  update target network
16:  repeat training epochs do
17:    get batch size experiences
18:    update network parameters
19:  end repeat
20: end for

```

Algorithm 2: Learning Algorithm using time series data

```

1: for episode = 1 to  $N$  do
2:   while step < max_step do
3:     get traffic states and waiting_time:  $s_t, w_t$ 
4:     estimate reward :  $r_t$ 
5:     get hidden state :  $h_{t-2}$ 
6:     que and pop time step states :  $ls_t$ 
7:     if  $ls_t > (max\_ls\_size) - 1$  then
8:       if len( $M$ ) > max_size - 1 then
9:         delete  $M[0]$ 
10:      end if
11:      append
12:      experience = ( $ls_{t-1}, a_{t-1}, r_t, ls_t, h_t$ )
13:      select action with  $\epsilon$ -greedy :  $a_t$ 
14:    else
15:      select action randomly
16:    end if
17:    yellow phase and green phase
18:     $t = t + 1$ 
19:    update  $ls_{t-1} = ls_t, a_{t-1} = a_t$ 
20:  end while
21:  repeat training epochs do
22:    get batch size experiences
23:    burn in process
24:    update network parameters
25:  end repeat
26: end for

```

shown in Figure 4 was prepared for simulation and learning. In this environment, vehicles and pedestrians travel on separate roads and do not collide at any point on the intersection except at crosswalks. During the simulation, the number of vehicles and pedestrians in an episode is determined at the stage where an episode is executed. At each step, vehicles and pedestrians are generated according to a certain probability and start moving to the destination determined

at the same time. Vehicles and pedestrians have their generation probabilities changed at certain steps. The purpose of this is to verify whether traffic signals are capable of responding to changes in traffic volume by establishing time periods during which traffic volume varies greatly.

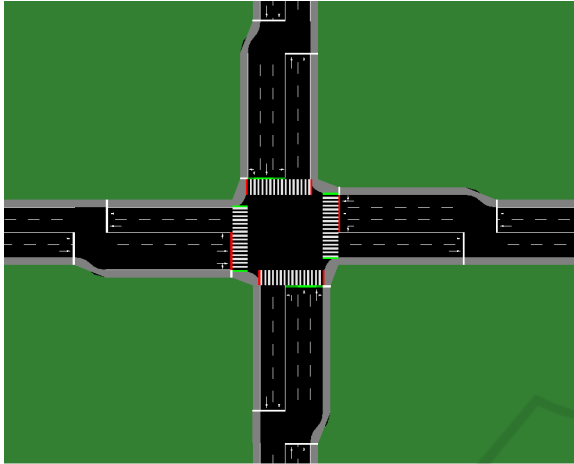


Figure 4: Simulation Environment.

4.2 Evaluation Experiment

4.2.1 Experimental Setup

To evaluate the performance of the trained traffic light agents, simulations up to 100 episodes were conducted. The evaluation was based on comparisons of average vehicle and pedestrian waiting times. In the experiment, traffic patterns with one point one times, one point five times, and two times the number of pedestrians as compared to vehicles are prepared, the traffic patterns are referred to as low-density, medium-density, and high-density traffic environments, respectively, and are evaluated. The performance was evaluated by comparing the learned traffic lights, DTC (DQN Traffic Control signal) and LTC (LSTM Traffic Control signal), with a traffic light called FTC (Fixed Traffic Control signal). DTC is a traffic light with a network of fully connected layers, and controls using the state of the intersection for one step at the intersection. LTC is a traffic light with a network including LSTM, and controls using the state for several steps at the intersection. FTC is a traffic light that switches color in a predetermined order at regular intervals and provides stable control regardless of the traffic environment. In this experiment, its signal is switched between the north-south pattern of P_0, P_2 and the east-west pattern of P_1, P_3 at 50 seconds intervals, as shown Section 3.4.

4.2.2 Experimental Result

Table 2: Vehicle waiting time.

Method	Low $\times 10^2 s$	Medium $\times 10^2 s$	High $\times 10^2 s$
<i>DTC</i>	25.7	29.4	29.5
<i>LTC</i>	248.9	348.2	520.3
<i>FTC</i>	106.5	107.0	111.2

Table 3: Pedestrian waiting time.

Method	Low $\times 10^2 s$	Medium $\times 10^2 s$	High $\times 10^2 s$
<i>DTC</i>	99.0	161.5	266.6
<i>LTC</i>	44.3	63.2	98.0
<i>FTC</i>	37.6	53.7	81.4

Table 4: Waiting time per vehicle.

Method	Low s	Medium s	High s
<i>DTC</i>	6.60	7.59	7.62
<i>LTC</i>	64.4	89.8	134.0
<i>FTC</i>	27.44	27.67	28.74

Table 5: Waiting time per pedestrian.

Method	Low s	Medium s	High s
<i>DTC</i>	20.80	25.04	27.68
<i>LTC</i>	9.35	9.49	9.96
<i>FTC</i>	8.05	8.26	8.46

Tables 2 through 5 show the experimental results. Tables 2 and 3 summarize the averaged cumulative values of waiting time per episode for vehicles and pedestrians. Table 4 shows the total amount of time vehicles wait between episodes divided by the number of vehicles, which is the waiting time per vehicle value. Table 5 shows the waiting time per pedestrian. The waiting time represents the time on the simulator, and the smaller the value is, the more appropriate the traffic control is for the environment.

4.2.3 Discussion

In Tables 2 to 5, which are the experimental results, we evaluate and discuss the accuracy of the learned traffic signals. Table 2 shows that, compared to the stable control of FTC, the learned traffic lights, DTC, reduced waiting time for vehicle by about 74% and increased vehicle waiting time for pedestrians by about 196%, respectively, on average at each density compared to FTC. Based on the wait times on both sides, DTC's overall wait time total is up. However, Table 4 and Table 5 shows that, compared to FTC, DTC, reduced the total waiting time of per vehicle and per pedestrians by about 12%. This can be thought of as a reduction in the amount of waiting time incurred by vehicles and pedestrians in the transportation network as a whole. Table 3 shows that for the two traffic signals trained, the waiting time for LTC is shorter, but Table 2 shows that the waiting time of vehicles at LTC is significantly higher than that at DTC. This is due to the fact that LTC has placed more emphasis on reducing pedestrian waiting time. From Section 3.4, the importance of the walker in the environment is adjusted by setting a constant α in the reward function. This high value the pedestrian learns that continuing to take actions that reduce the pedestrian's waiting time is a simple way to increase the value of the reward. Therefore, it is necessary to design rewards that reduce waiting time for both pedestrians and vehicles. It can also be inferred that the presence of the prepared road environment was a factor that increased the overall waiting time including vehicles. Since the prepared environment is only a crossroad, it can be said that it is easy to grasp the scale of vehicles and pedestrians moving in the road. In such an environment, it can be judged that DTC, which uses the current state of the environment, is easier to perform control that reflects the state of the environment. On the other hand, LTC can lead to better control in complex road environments where it is difficult to judge traffic conditions on the spot. Tables 4 and 5 show that DTC and FTC does not show a significant change in waiting time per vehicle or per pedestrian with respect to changes in traffic density. In contrast, the learned traffic signals show a large change in vehicle waiting times, but no noticeable difference in pedestrian waiting times. This indicates that the learned traffic signals work to reduce the waiting time for pedestrians in response to changes in the traffic environment. Therefore, it can be determined that the learned traffic signals are superior in terms of control adapted to the traffic environment.

5 CONCLUSION

The signal condition was verified by using a network that uses the state of the environment between one step and a network that uses the state of the environment between several steps. In this research, traffic control is performed using traffic signals learned by deep reinforcement learning in an environment with a mixture of vehicles and pedestrians. As a result, DTC, a learned traffic light, led to a reduction in the waiting time experienced by each of the traffic network as a whole. Furthermore, LTC was not possible to reduce the waiting time of the entire traffic network, but it was possible to reduce the pedestrian waiting time by adapting to changes in traffic volume. In the future, we will expand the learning and experimental environment to create a traffic signal system that can control traffic in a large-scale traffic network. In addition, we will improve the control of traffic signals using time series information. We will try to control appropriately by increasing or decreasing the number of states obtained from the environment and by adjusting the past states used to select actions and adjusting the past states used to select actions. On top of that, we try to control not only simple structures such as crossroads, but also complex road environments by using human flow, such as the size of pedestrian groups and the direction of movement.

ACKNOWLEDGEMENTS

This work was supported by JSPS KAKENHI Grant Numbers JP21H03496, JP22K12157.

REFERENCES

- Behrisch, M., Lopez, P. A., Bieker-Walz, L., Erdmann, J., Flotterod, Y., Hilbrich, R., Lucken, L., Rummel, J., Wagner, P., and WieBner, E. (2018). *Microscopic Traffic Simulation using SUMO*. IEEE Intelligent Transportation Systems Conference (ITSC).
- Choe, C., S.Back, Woon, B., and Kong, S. (2018). *Deep Q Learning with LSTM for Traffic Light Control*. 2018 24th Asia-Pacific Conference on Communications(APCC).
- Ge, H., Y.Song, Wu, C., Ren, J., and Tan, G. (2019). *Co-operative Deep Q-Learning With Q-Value Transfer for Multi Intersection Signal Control*. 2019 IEEE Access 2907618.
- Gers, F. A., Schmidhuber, J., and Cummins, F. (2000). *Learning to forget: continual prediction with LSTM*. Neural Computation 12(10), 2451-2471.

- Hausknecht, M. and Stone, P. (2017). *Deep Recurrent Q-Learning for Partially Observable MDPs*. arXiv:1507.06527v4.
- Hochreiter, S. and Schmidhuber, J. (1997). *LONG SHORT-TERM MEMORY*. *Neural Computation* 9(8): 1735-1780.
- Kapturowski, S., Ostrovsk, G., Quan, J., and Dabney, R. M. W. (2019). *Recurrent Experience Replay In Distributed Reinforcement Learning*. ICLR 2019.
- Liang, X., X.Du, and Han, Z. (2018). *Deep reinforcement learning for traffic light control in vehicular networks*. *IEEE Trans. Veh. Technol.*: [https://arxiv.org/abs/1803.11115\(2022/06/10](https://arxiv.org/abs/1803.11115(2022/06/10) reference).
- Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, L., Wierstra, D., and Riedmiller, M. (2013). *Playing Atari with Deep Reinforcement Learning*. arXiv preprint arXiv:1312.5602.
- Vidali, A. (2017). *Simulation of a traffic light scenario controlled by Deep Reinforcement Learning agent*. [https://github.com/AndreaVidali/Deep-QLearning-Agent-for-TrafficSignal-Control\(2022/06/10](https://github.com/AndreaVidali/Deep-QLearning-Agent-for-TrafficSignal-Control(2022/06/10) reference).

