



Learning Thresholds to Select Cooperative Partners by Applying Deep Reinforcement Learning in Distributed Traffic Signal Control

Shinya Matsuta^{1*}, Naoki Kodama² and Taku Harada¹

¹Tokyo University of Science

²Meiji University

7420527@alumni.tus.ac.jp, kodama@meiji.ac.jp, harada@rs.tus.ac.jp

Abstract

One method to reduce vehicle congestion in a road traffic network is to appropriately control traffic signals. One control scheme for traffic signals is a distributed control scheme in which individual traffic signals cooperate locally with other geographically close traffic signals. Deep reinforcement learning has been actively studied to appropriately control traffic signals. In distributed control, it is important to select appropriate cooperative partners. In this study, we propose a method for selecting appropriate cooperative partners using deep reinforcement learning to the distributed traffic signal control.

1 Introduction

In recent years, road traffic networks have become congested because of the increasing volume of vehicular traffic. One method to alleviate traffic congestion is to appropriately control traffic signals. The control of traffic signals determines the time during which the same color, such as green, is continuously displayed in a traffic signal.

There are two main control methods for traffic signals: centralized and decentralized. In the centralized control method, all traffic signals in the entire road traffic network, such as a control center, are centrally controlled at a single location. However, there are some problems: the amount of computation required for controlling becomes enormous as the number of traffic signals increases, and if the control equipment at the control center malfunctions, not all traffic signals can be properly controlled.

In contrast, in the decentralized control scheme, individual traffic signals are controlled autonomously by locally coordinating traffic signals located in close geographic proximity to each other.

* Affiliation as of March 2022

In this control scheme, even if an individual traffic signal malfunctions, its impact on the entire road traffic network is small.

Reinforcement learning, a machine learning algorithm, has been actively studied for distributed traffic signal control [1]-[3]. Using reinforcement learning, optimal control can be learned by considering the interaction with the environment surrounding traffic signals. In many studies on the distributed traffic signal control using reinforcement learning, each traffic signal cooperates with all adjacent traffic signals. However, it is common for traffic volumes on each road to vary; therefore, rather than coordinating with all adjacent traffic signals, it may be possible to achieve more appropriate control by limiting the coordination partners to a portion of the traffic signals, based on the traffic volumes and other factors. Furthermore, the amount of information used for coordination can be reduced by limiting coordination partners to a part of traffic signals.

Deep reinforcement learning is a machine learning algorithm that combines a neural network with a reinforcement learning algorithm. In this study, we propose a learning method for the distributed traffic signal control using deep reinforcement learning, in which the cooperative partners are limited to a subset of traffic signals instead of cooperating with all adjacent traffic signals, and evaluate the effectiveness of the learning method. We have previously studied the distributed traffic signal control using deep reinforcement learning [4]-[6]. In [4], our proposed deep reinforcement learning algorithm was used to reduce the waiting time of vehicles in an entire road traffic network. In addition, in [5][6], we proposed a method to limit cooperative partners to a part of the network. In [5], the cooperative partner was predetermined. In [6], the cooperative partner was not determined in advance, but a parameter used for selecting the cooperative partner was defined, and the value of this parameter was set in advance. In contrast, in this study, the values of the parameters predetermined in [6] were obtained by learning. In this study, deep Q-network (DQN) [7] was applied as a specific deep reinforcement learning algorithm. Furthermore, we targeted a mesh-like road environment as the road traffic network geometry.

2 Related research

Ge et al. proposed a cooperative deep-Q network by transferring the Q values for adaptive road traffic signal control [8]. Kamoto et al. proposed a method to identify a set of intersections that must be considered for interaction [9][10]. However, the effectiveness of the proposed method has not yet been fully evaluated. Wang et al. used a graph attention network to recognize different neighborhood agents and assigned different weights to nearby intersections [11]. Jiang et al. proposed a model that decomposed intersections into subgraphs such that subgraphs, rather than entire graphs, were learned synchronously, thereby significantly reducing the learning time [12]. Su et al. proposed a multiagent reinforcement learning algorithm combined with an attention mechanism for a large-scale traffic signal control problem [13]. Zeng extracted the features of dynamic traffic networks using graph convolutional networks and used the states of neighboring A2C agents to learn cooperative control policies [14]. Wang et al. proposed a new multiagent reinforcement learning method called cooperative dual-Q learning (Co-QL), which was applied to traffic signal control and tested on various traffic flow scenarios. [15]. Du et al. used graph neural networks to achieve multiagent coordination [16]. Chu et al. proposed a novel A2C-based multiagent reinforcement learning algorithm for scalable and robust adaptive road traffic light control [17]. Zhang et al. modeled an adaptive traffic light control problem as a networked Markov game. They then analyzed the degree of correlation between connected neighbors, weight observations, and rewards based on this degree of correlation [18]. Graves et al. described a decentralized road traffic light control structure that performed cooperative traffic light control through repeated negotiations with neighboring agents [19]. Haddad et al. proposed an approach to integrate cooperative activities among agents so that they could share their decisions and observations [20]. Yang

et al. proposed an algorithm for traffic light control based on current traffic conditions and past observations [21]. Zhao et al. proposed a reinforcement learning-based method that could rapidly realize control measures to maximally reduce the average vehicle travel time [22]. Elise van der Pol introduced variable elimination and max-plus, a coordination algorithm applicable to cooperative multi-agent systems represented by a coordination graph [23]. However, for the distributed control of road traffic signals, there is little research on the proper selection of a partner road traffic signal to cooperate with.

We previously studied the application of deep reinforcement learning to decentralized control of traffic signals [5][6]. In [5], we showed that the average speed of the entire road network can be improved by coordinating only with traffic signals in the direction of heavy traffic. However, in this study, the traffic signals with which to cooperate are predefined. In [6], a threshold value was set for the percentage of traffic entering an intersection, and only traffic signals in the direction greater than the threshold value were coordinated. However, this threshold value was set in advance based on preliminary experiments. Therefore, several preliminary experiments are required to set an appropriate threshold value.

Based on the decentralized control method for traffic signals proposed in literature [6], this study proposes a method that uses traffic density as an evaluation index to select traffic signals to cooperate, and also learns the threshold value, which is the boundary line between cooperating and not cooperating.

3 Deep Q-Network

Reinforcement learning is a machine-learning algorithm in which an agent acts on its environment and seeks strategies to maximize the rewards it receives. Strategy is a rule that serves as a measure of an agent's action. Reward is a measure of the goodness of the agent's action and the state of the environment, whereas the agent modifies the action strategy based on the reward. In reinforcement learning, the following steps are repeated: "the agent recognizes the state of the environment," "the agent selects and executes an action," "the state of the environment is updated by the action executed by the agent, and the agent receives a reward for the action or the state in some cases," and "the agent modifies the strategy for the action based on the reward".

Q-learning is a specific reinforcement learning algorithm, which maintains a tabular representation of the Q-value of an agent's actions, and updates the Q-value for each action. The Q-value was calculated as the expected value of the discounted return of the possible future reward. DQN [7] is an extension of Q-learning and combines Q-learning with neural networks. Specifically, a neural network is used to approximate the Q-value of each action with respect to the state of the environment.

$$Q(s, a, \theta) \leftarrow R(s, a) + \gamma \max_{a'} Q(s', a', \bar{\theta}), \quad (1)$$

where Q represents the Q-value, s represents the state, s' represents the next state, γ represents the current value of the future reward, a represents the action, R represents the reward, and θ represents the training parameter of the neural network. $\bar{\theta}$ represents the parameters of the target network [7].

4 Proposed method for selecting cooperative partners

4.1 Agent design

The traffic signal at an intersection is represented by a single agent. The traffic signal agent perceives the state, chooses an action, and receives a reward from the environment based on the result of the action taken.

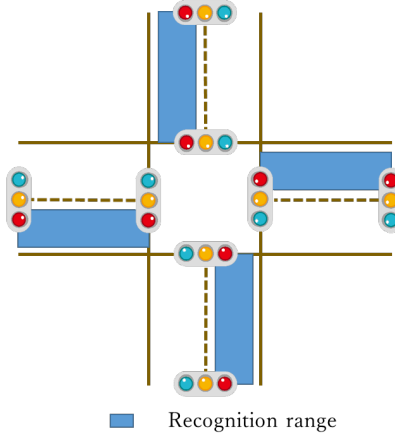


Figure 1: Range of roads to be recognized

In this study, a mesh road environment was targeted. There are two types of traffic signal phases: "north-south green light and east-west red light" and "north-south red light and east-west green light." The range of roads recognized by each traffic signal agent is shown in Figure 1. The traffic signal agent recognizes the following five states [5][6]: its own phase within a one-step range, the duration of that phase [s], traffic density [vehicles/km], number of waiting vehicles [vehicles], and average speed [km/h]. The duration of a phase is the time that a "green light on north-south and red light on east-west" or a "red light on north-south and green light on east-west" condition is maintained. Traffic density, number of vehicles waiting, and average speed are the traffic density, number of vehicles stopped, and average speed on the road in each of the four directions flowing into itself, respectively. The phase duration, traffic density, and the number of vehicles waiting were normalized to consider values between 0 and 1, facilitating their use as state inputs to the agent [5][6]. For the average speed, we used the value of the average speed v [km/h] of all vehicles within the recognition range, normalized to a value between 0 and 1. For this normalization, the observed speed was divided by 120 to obtain a range between 0 and 1, as the speed can consider values between 0 km/h and slightly over 100 [km/h]. Note that when switching phases, a yellow signal phase of a fixed duration is inserted, regardless of the intention of the traffic signal agent.

In this study, the average speed, which was allowed to range from 0 to approximately 1, was used as the reward R [5][6]. The reward is the average velocity set to range from 0 to approximately 1. This is shown in Equations (2) and (3) as follows.

$$R = v' \quad (2)$$

$$v' = \frac{v}{120} \quad (3)$$

4.2 Methods of selecting cooperative partners

Previously, we adopted a method that only works with traffic signals with a high volume of incoming traffic to themselves and adjacent traffic signals by introducing a traffic volume threshold [5][6]. This study proposes a method to obtain the threshold value for selecting a cooperative partner through learning. In other words, in addition to the two phases, the threshold value is learned. In this study, the threshold value is related to traffic density.

Because DQN considers only discrete values of action, in this study, we prepared 6%, 7%, 8%, 9%, and 10% as possible threshold values based on the results of preliminary experiments. The appropriate threshold value was determined by learning from



Action	Signal colors in north-south direction	Threshold
1		6%
2		7%
3		8%
4		9%
5		10%
6		6%
7		7%
8		8%
9		9%
10		10%

Table 1: Types of action

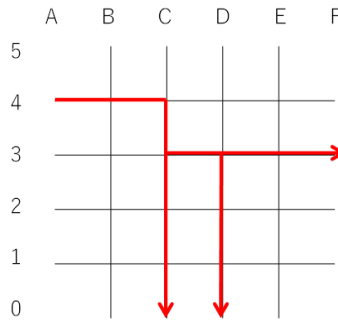


Figure 2: Road traffic network

among these values. That is, there are a total of 10 types of actions: "north-south red light, east-west green light, 6% threshold," "north-south red light, east-west green light, 7% threshold," ..., "north-south blue light, east-west red light, 10% threshold". The types of action are listed in Table 1.

Let n be the maximum number of steps in the learning session. Then, the timing for selecting an action during one learning session is every steps-step. That is, one of the actions listed in Table 1 is selected at each s -step. Therefore, the color of the display may change based on the selected action when the action is selected, and the minimum duration of the display color is s steps. The traffic density is calculated using the driving conditions of the vehicles when up to n steps are executed. If there is no road with traffic density exceeding the threshold on the incoming road, the system coordinates with all traffic signals on the upper, lower, left, and right sides of the road. Furthermore, in the first run, the system cooperated with all the traffic signals on the upper, lower, left, and right sides.

5 Evaluation experiment

5.1 Experimental environment

In this experiment, we used a 4×4 mesh road traffic network with a distance of 400 m between the intersection and edge of the environment as shown in Figure 2. This indicates that the length of one straight road from one end of the environment to the other was 2 km.

Vehicles appeared with a certain probability at each step from the edge of the environment and disappear when they arrive at a different edge from where they appear. In this experiment, two types of vehicle appearance rates were used, and the vehicles that appeared at each rate were referred to as vehicle A and vehicle B. First, vehicle A appears from all environmental edges with a probability of 0.0011 per step. Second, vehicle B chooses the path indicated by the arrow in Figure 1 and appears with a probability of 0.9 per step. By introducing vehicle B, we aim to increase congestion on some routes. When traveling to a destination, the vehicle chooses the route that requires the least time to reach the destination. Therefore, it may choose a route that requires a detour to reach the destination.

In Figure 2, A-F denote columns and 0-5 denote rows. Each intersection is denoted by A1, A2, A3, ..., and F5 using the row and column symbols, respectively. Arrows indicate roads with high traffic volume. The three roads with the highest traffic volumes were set as follows.

- (1) Roads first pass through the traffic signal at intersection B4 and then through intersections C4, C3, C2, and C1.
- (2) Roads first pass through the traffic signal at intersection B4 and then through intersections C4, C3, D3, and E3.
- (3) The road first passes through a traffic signal at intersection B4 and then through intersections C4, C3, D3, D2, and D1.

Note that 90% of the vehicles passing through intersection D3 proceeded to intersection E3, whereas 10% proceeded to intersection D2 and through intersection D1. Roads are considered to be ordinary roads; however, considering their application in an international road traffic network, the speed limit on ordinary roads is considered to be 90 km/h. Vehicles can perform four types of movements at each intersection: straight, left-turn, right-turn, and U-turn. The total number of signal agents is 16. The maximum time that a traffic signal agent can continue the red and blue signal phases is 50 s and the minimum time is 5 s. When switching phases, a yellow signal phase of 2 s is always inserted, regardless of the signal agent's intention. The simulation environment was the micro traffic flow simulator Simulation of Urban MObility (SUMO) [23], an open-source traffic simulator that provides an API and GUI. In this experiment, one step of the agent's interaction with the environment was defined as 5 s of time in the simulation.

5.2 Experimental scenario

The following three experimental scenarios will be set:

[Scenario 1]

Every traffic signal agent perceives the state of all roads adjacent to its own intersection. In other words, each agent can cooperate with adjacent four-way traffic signal agent.

[Scenario 2]

Each traffic signal agent recognizes its own intersection and the state of the roads flowing into it whose traffic density exceeds a predefined threshold to be able to cooperate only with traffic signal agents whose congestion exceeds the threshold. In our experiments, we used five predefined thresholds: 6%, 7%, 8%, 9%, and 10%.

[Scenario 3]

This was the proposed method that performs threshold learning and its utilization. In utilization, the system detects intersections and road conditions that exceed the threshold values obtained through learning, and cooperates with traffic signal agents in that direction. Before the experiment, we prepared five threshold values (6%, 7%, 8%, 9%, and 10%). We learned the appropriate threshold value from among multiple threshold value prepared in advance.

5.3 Parameter setting

In this experiment, we used a 3-layer all-coupled layer consisting of 256 nodes as the DQN network architecture. The inputs were four-dimensional, and the outputs are two-dimensional when learning the threshold for selecting a cooperative partner proposed in this study, and one-dimensional otherwise. Adam [24] was used to optimize the neural network, and the parameter settings followed Adam's literature [24]. The ϵ -greedy method was employed to select actions. ϵ linearly decreased from the initial value of 1 to 0.02 after 10000 steps. The remaining parameters followed those in literature [7]. After an action was selected, the selected traffic signal phase was assumed to continue for 5 s. However, if a yellow signal was included, it continued with yellow signal phase 2 s + selected traffic signal phase 3 s. One run was set up as a 10,000 s simulation. To wait for enough vehicles to enter the road network, learning began after 500 s. This procedure was repeated 1,000 times. For the evaluation, we tested the trained model after 10,000 s and used the average speed of all the vehicles in the environment at that time.

5.4 Experimental results and discussion

Scenarios 1, 2, and 3 were run 10 times each. Table 2 lists the experimental results for Scenario 2. Table 2 shows that among the threshold values fixed in advance in the Scenario 2 experiment, the average speed was the highest when the threshold value was 8%. Furthermore, it can be observed that the threshold value affects the variation in the average speed and running speed from experiment to experiment.

The results of the 10 evaluations for Scenarios 1, 2 (threshold of 8%), and 3 are shown in Table 3. The average speeds are listed in Table 3. Table 3 shows that the proposed Scenario 3 is slightly higher speed than Scenario 1, in which all adjacent traffic signals are coordinated. However, when comparing Scenario 2, which uses a fixed threshold, and Scenario 3, which uses the proposed method, the results for Scenario 2 were slightly higher speed than those for Scenario 3. However, in the setting of this study, neither result was significantly different from the other. However, the results of Scenario 2 were the result of an experiment with an appropriate threshold (8%) obtained through a preliminary experiment. Subsequently, we discuss the standard deviation.

Table 3 shows that compared to the case where the threshold value was fixed in advance, the proposed method resulted in smaller standard deviation. However, when compared to the case where the threshold value was fixed at 8%, which had the highest average speed, and the mean value (0.58) of the standard deviation for the five threshold values, the standard deviation for the proposed method was small. The reason for the smaller standard deviation in the proposed method (Scenario 3) compared with the standard deviation at the 8% threshold, which had the highest average speed, can be attributed to the effect of learning the threshold value, which is the proposed method. In Scenario 2, the threshold was set to the same value for all intersections. In contrast, the proposed method (Scenario 3) allowed the selection of a different threshold value for each intersection. In the experiment, the average threshold value in all steps for all intersections in Scenario 3 was almost the same as that in Scenario 2 (8%); however, in each step, each intersection was free to choose a threshold value between 6% and 10%, which is a candidate threshold value. The experimental results for Scenario 2 showed that when the threshold was set to the same value at all intersections, the 8% threshold resulted in the highest average speed; however, there was only a slight difference in the average speed from the 9% threshold, and the standard deviation was smaller for the 9% threshold. Therefore, by learning the threshold value itself, it is possible to freely select the threshold value based on the traffic volume, which may have led to more stable learning compared with the case where a fixed threshold value of 8% was always applied. However, it has not been shown that the threshold value at each intersection is appropriately selected based on the traffic volume.

	Scenario2(6%)	Scenario2(7%)	Scenario2(8%)	Scenario2(9%)	Scenario2(10%)
1	91.44	90.09	90.21	91.63	91.50
2	91.01	91.21	90.57	91.08	90.39
3	91.93	90.69	91.45	91.38	91.49
4	90.14	91.38	91.20	91.46	91.01
5	91.06	90.31	91.11	91.13	91.55
6	90.81	91.22	91.27	90.22	91.01
7	91.67	90.99	91.20	90.77	89.05
8	91.01	91.10	91.52	91.44	90.49
9	91.18	91.10	89.99	90.87	90.41
10	89.63	90.73	91.54	91.04	91.74
Mean	90.99	90.88	91.11	91.10	90.86
Standard Deviation	0.68	0.42	0.56	0.41	0.81

Table 2: Comparison of average speeds in preliminary experiments for Scenario 2

	Scenario 1	Scenario 2 (8%)	Scenario 3
1	91.32	90.21	91.20
2	91.13	91.57	91.31
3	90.72	91.45	91.21
4	89.66	91.20	91.37
5	91.55	91.11	90.96
6	90.67	91.27	91.33
7	90.98	91.20	89.98
8	90.48	91.52	90.86
9	91.40	89.99	91.77
10	90.31	91.54	90.86
Mean	90.82	91.11	91.09
Standard Deviation	0.57	0.56	0.47

Table 3: Comparison of average speeds for each scenario

6 Conclusion

In this study, we proposed a method to obtain the value of the threshold for cooperating only with traffic signals in a lane where the traffic density of vehicles entering an intersection exceeded the threshold in the decentralized control of traffic signals using deep reinforcement learning through learning. To evaluate the effectiveness of the proposed method, we conducted experiments on a 4×4 mesh road traffic network. The results showed that the proposed method was slightly faster in terms of travel speed than in the case in which the proposed method cooperated with all adjacent traffic signals. But, the proposed method was slower than the case in which the threshold value was fixed in advance at the optimal value obtained from the preliminary experiments. However, in the current experiment, the proposed method did not show any significant differences in either case.

On the other hand, the standard deviation of the running speed per number of experiments was relatively small for the proposed method compared to the case where the proposed method cooperated

with all adjacent traffic signals. Compared to the case where we fixed the threshold value in advance, the proposed method produced a lower running speed in some cases and larger running speed in others. However, compared with the case where the threshold value was fixed at 8%, which had the highest average speed, and the average of the standard deviations for the five threshold values, the standard deviations for the proposed method were small. This indicates that the proposed method is relatively stable for learning. For a more detailed comparison, verification by more runs than the 10 runs performed in this experiment is required.

Future work will include updating the state definitions and improving the proposed method so that sufficient significant differences can be obtained with respect to speed.

References

- [1] H. Wei, G. Zheng, V. Gayah and Z. Li, "Recent Advances in Reinforcement Learning for Traffic Signal Control," *A Survey of Models and Evaluation, ACM SIGKDD Explorations Newsletter*, vol. 22, no. 2, pp. 12-18, 2020.
- [2] M. Miletic, E. Ivanjko, M. Greguric and K. Kusic, "A review of reinforcement learning applications in adaptive traffic signal control," *IET Intelligent Transport Systems*, vol. 16, no. 10, pp. 1269-1285, 2022.
- [3] M. Noaen, A. Naik, L. Goodman, J. Crebo, T. Abrar, Z. S. H. Abad, A. L. Bazzan and B. Far, "Reinforcement learning in urban network traffic signal control," *A systematic literature review; Expert Systems With Applications*, vol. 199, 2022.
- [4] N. Kodama, T. Harada and K. Miyazaki, "Traffic Signal Control System Using Deep Reinforcement Learning with Emphasis on Reinforcing Successful Experiences," *IEEE Access*, vol. 10, pp. 128943-128950, 2022.
- [5] S. Matsuta, N. Kodama and T. Harada, "Fundamental Study on Distributed Control of Road Traffic Signals," *IEEJ System Study Group*, Vols. ST-21-012, pp. 5-8, 2021 (in Japanese).
- [6] S. Matsuta, N. Kodama and T. Harada, "Proposal for selecting a cooperation partner in distributed control of traffic signals using deep reinforcement learning," *Proceedings of the 8th IIAE International Conference on Intelligent Systems and Image Processing 2021*, pp. 153-158, 2021.
- [7] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg and D. Hassabis, "Human-level control through deep reinforcement learning," *NATURE*, vol. 518, pp. 529-533, 2015.
- [8] H. Ge, Y. Song, C. Wu, J. Ren and G. Tan, "Cooperative Deep Q-Learning With Q-Value Transfer for Multi-Intersection Signal Control," *IEEE Access*, pp. 40797-40809, 2019.
- [9] M. Kamon and S. Arai, "Evaluation of Information Sharing Effects for Cooperative Control of Traffic Signals," *The 23rd National Conference of Japanese Society for Artificial Intelligence*, 2009.
- [10] M. Kamon and S. Arai, "Group-specific traffic light control by extracting intersection sets with dense relationship using traffic network structure and traffic flow," *Conference on Systems and Information 2009 of the Society of Instrument and Control Engineers*, pp. 568-571, 2009.

- [11] M. Wang, L. Wu, J. Li and L. He, "Traffic Signal Control With Reinforcement Learning Based on Region-Aware Cooperative Strategy," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 6774-6785, 2022.
- [12] S. Jiang, Y. Huang, M. Jafari and M. Jalayer, "A Distributed Multi-Agent Reinforcement Learning With Graph Decomposition Approach for Large-Scale Adaptive Traffic Signal Control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 14689-14701, 2022.
- [13] C. Su, Y. Yan, T. Wang, B. Zhang and C. Li, "A Graph Attention Mechanism Based Multi-Agent Reinforcement Learning Method for Efficient Traffic Light Control," *2021 International Wireless Communications and Mobile Computing*, pp. 1332-1337, 2021.
- [14] Z. Zeng, "GraphLight: Graph-based Reinforcement Learning for Traffic Signal Control," *2021 IEEE the 6th International Conference on Computer and Communication Systems*, pp. 645-650, 2021.
- [15] X. Wang, L. Ke, Z. Qiao and X. Chai, "Large-Scale Traffic Signal Control Using a Novel Multiagent Reinforcement Learning," *IEEE Transactions on Cybernetics*, vol. 51, no. 1, pp. 174-184, 2021.
- [16] X. Du, J. Wang, S. Chen and Z. Liu, "Multi-agent Deep Reinforcement Learning with Spatio-Temporal Feature Fusion for Traffic Signal Control; Joint European Conference on Machine Learning and Knowledge Discovery in Databases," *ECML PKDD 2021: Machine Learning and Knowledge Discovery Applied Data Science Track*, pp. 470-485, 2021.
- [17] T. Chu, J. Wang, L. Codeca and Z. Li, "Multi-Agent Deep Reinforcement Learning for Large-Scale Traffic Signal Control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 1086-1095, 2020.
- [18] C. Zhang, Y. Tian, Z. Zhang, W. Xue, X. Xie, T. Yang, X. Ge and R. Chen, "Neighborhood Cooperative Multiagent Reinforcement Learning for Adaptive Traffic Signal Control in Epidemic Regions," *IEEE Transactions on Intelligent Transportation Systems*, 2022.
- [19] R. T. Graves, Z. E. Nelson and S. Chakraborty, "A decentralized intersection management system through collaborative negotiation between smart signals," *Journal of Intelligent Transportation Systems*, 2021.
- [20] T. A. Haddad, D. Hedjazi and S. Aouag, "A deep reinforcement learning-based cooperative approach for multi-intersection traffic signal control control," *Engineering Applications of Artificial Intelligence*, vol. 114, 2022.
- [21] S. Yang, B. Yang, H.-S. Wong and Z. Kang, "Cooperative traffic signal control using Multi-step return and Off-policy Asynchronous Advantage Actor-Critic Graph algorithm," *Knowledge-Based Systems*, vol. 183, 2019.
- [22] W. Zhao, Y. Ye, J. Ding, T. Wang, T. Wei and M. Chen, "IPDALight: Intensity- and phase duration-aware traffic signal control based on Reinforcement Learning," *Journal of Systems Architecture*, vol. 123, 2022.
- [23] E. v. d. Pol, "Deep Reinforcement Learning for Coordination in Traffic Light Control," *Master Thesis, University of Amsterdam*, 2016.
- [24] D. Krajzewicz, G. Hertkorn, C. Feld and P. Wagner, "SUMO (Simulation of Urban MObility); An open-source traffic simulation," *4th Middle East Symposium on Simulation and Modelling*, 2002.
- [25] D. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *Proceeding of The International Conference on Learning Representations 2015*, 2015.