Micro Junction Agent: A Scalable Multi-agent Reinforcement Learning Method for Traffic Control

BumKyu Choi[a], Jean Seong Bjorn Choe[b] and Jong-kook Kim
School of Electrical Engineering, Korea University, An-am Street, Seoul, South Korea

Keywords: Multi-agent, Reinforcement Learning, Traffic Control, Traffic Intersection Management.

Abstract: Traffic congestion is increasing steadily worldwide and many researchers have attempted to employ smart methods to control the traffic. One such approach is the multi-agent reinforcement learning (MARL) scheme wherein each agent corresponds to a moving entity such as vehicles. The aim is to make all mobile objects arrive at their target destination in the least amount of time without collision. However, as the number of vehicles increases, the computational complexity increases, and therefore computation cost increases, and scalability cannot be guaranteed. In this paper, we propose a novel approach using MARL, where the traffic junction becomes the agent. Each traffic junction is composed of four Micro Junction Agents (MJAs) and a MJA becomes the observer and the agent controlling all vehicles within the observation area. Results show that MJA outperforms other MARL techniques on various traffic junction scenarios.

1 INTRODUCTION

Traffic congestion in cities is getting severe as the metropolitan area expands. Traffic congestion triggers harmful effects on the environment and low energy efficiency of transportation. Various social studies show that Americans burn approximately 5.6 billion gallons of fuel each year simply idling their engines (Lasley, 2019; Fiori et al., 2019). Recent advances in autonomous vehicles, vehicle to vehicle communication, and Internet of Things (IoT) infrastructures will allow the control of vehicles to alleviate traffic congestion. Researchers in various fields attempted to solve the traffic congestion problem, by applying optimization theories and heuristic techniques (Hartanti et al., 2019; Khoza et al., 2020). As machine learning methods became successful in many fields of research, reinforcement learning has been applied to traffic control (Walraven et al., 2016).

There are two major approaches to traffic control. One major approach is traffic signal control (Wei et al., 2019) and is extensively studied. The traffic congestion problem is solved by intelligently controlling the traffic light system. Many schemes such as rule-based, heuristic method, and multi-agent methods are used. However, traffic lights have been argued as a suboptimal way of managing intersections (Dresner and Stone, 2008). Another major branch of research is the traffic junction or intersection problem where there are no traffic lights (Figure 1). Many researchers use a multi-agent method to solve this problem. A machine learning method called multi-agent reinforcement learning (MARL) is used (Buşoniu et al., 2010) to enhance the performance. The absence of traffic lights proves to be a more difficult problem but if the vehicles are controlled correctly, the traffic junction can be a better system to reduce the traffic congestion. For all traffic control research, the collision of the vehicles means failure. Thus, the goal will be the intelligent control of the overall system to guide the vehicles to their predetermined destination while avoiding collisions. The environment in this paper assumes an autonomous intersection management problem (Dresner and Stone, 2008) and all vehicles' routes are predetermined.

We propose a novel method to tackle the no traffic light traffic junction problem for autonomous vehicles. Our method combines two key ideas. The first is that a junction or intersection is partitioned into homogeneous Micro Junction Agents (MJAs). Each MJA controls the vehicles that are situated in the MJA's predetermined governing area. Second, the intersection management problem is formulated as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP) (Bernstein et al., 2002) and

[a] https://orcid.org/0000-0002-5327-4502
[b] https://orcid.org/0000-0002-4865-5116

Choi, B., Choe, J. and Kim, J.
Micro Junction Agent: A Scalable Multi-agent Reinforcement Learning Method for Traffic Control.
DOI: 10.5220/0010849600003116
In *Proceedings of the 14th International Conference on Agents and Artificial Intelligence (ICAART 2022) - Volume 3*, pages 509-515
ISBN: 978-989-758-547-0; ISSN: 2184-433X

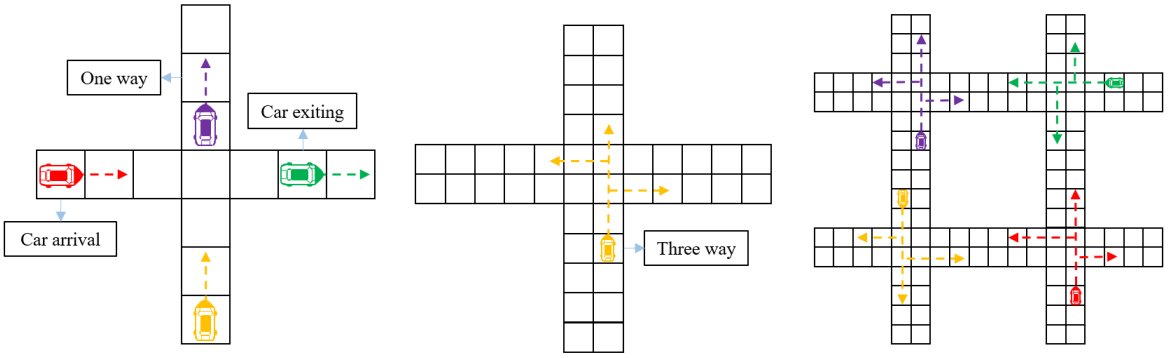


Figure 1: Traffic junction environment and an example of vehicle movements in three modes.

solved using a version of simple parameter-shared independent Q-learning (IQR) (Tan, 1993). By designing the observation space and the action space of a MJA to be optimized, we are able to build a scalable multi-agent traffic control system without explicit communication.

This paper is organized as follows. In Section 2, related research is introduced. Section 3, describes our proposed approach. In section 4, the experiments and results are evaluated and compared to other methods. Section 5 concludes this paper.

2 RELATED WORK

Several studies were conducted using the traffic junction environment (Sukhbaatar et al., 2016; Singh et al., 2018; Das et al., 2019; Su et al., 2020). These works are mainly about achieving a common goal of guiding vehicles to their destination without any collision by intelligently communicating between moving agents. These papers use reinforcement learning where explicit communication between vehicles under partial observability is assumed. Our approach requires communication between a traffic junction and vehicles within its observation boundary only. Therefore, the proposed method in this paper require fewer communication and the computation cost is a function of the number of traffic junctions not the vehicles.

The problem in this paper is closely related to a problem called autonomous intersection management (AIM), proposed in (Dresner and Stone, 2008), where each intersection intelligently controls vehicles that pass through. (Hausknecht et al., 2011) studies this problem when multiple intersections exist. This multi-intersection setup is almost identical to the traffic junction environment. The difference is that the traffic junction environment assumes that the route of each vehicle is pre-determined. Our approach introduces a multi-agent reinforcement learning scheme, where an intersection or a traffic junction is divided into four micro junction agents (MJAs). Contrary to the approaches using a heuristic method (Chouhan and Banda, 2018; Parker and Nitschke, 2017), our reinforcement learning scheme enables higher scalability.

3 PROPOSED APPROACH

3.1 Traffic Junction Environment

The traffic intersection environment introduced in CommNet (Sukhbaatar et al., 2016) is used for the experimental environment (Figure 1). The traffic junction is a two-dimensional discrete-time traffic simulation environment. Vehicles have pre-assigned routes and are randomly added to the traffic junctions with probability P_{arrive} at each time step. They occupy a single cell at any given time and have two possible actions: *gas* (go forward) and *stay* (stop). A vehicle will be removed once it reaches its destination grid cell.

The total number of vehicles at any given time is fixed at N_{max}. In each time-step, they communicate and need to avoid collisions with each other and reach their destinations. If all vehicles do not crash until max-steps, we treat it as a success for that episode. The task has three difficulty level settings (easy, medium, hard) (Figure 1), and each mode varies in the number of possible routes, junctions, and entry points.

3.2 Model

The traffic junction environment can be seen as a fully cooperative and partially observable multi-agent problem (Gupta et al., 2017). As a fully cooperative system, all agents obtains the global reward for every time step. We further assume to have homoge-

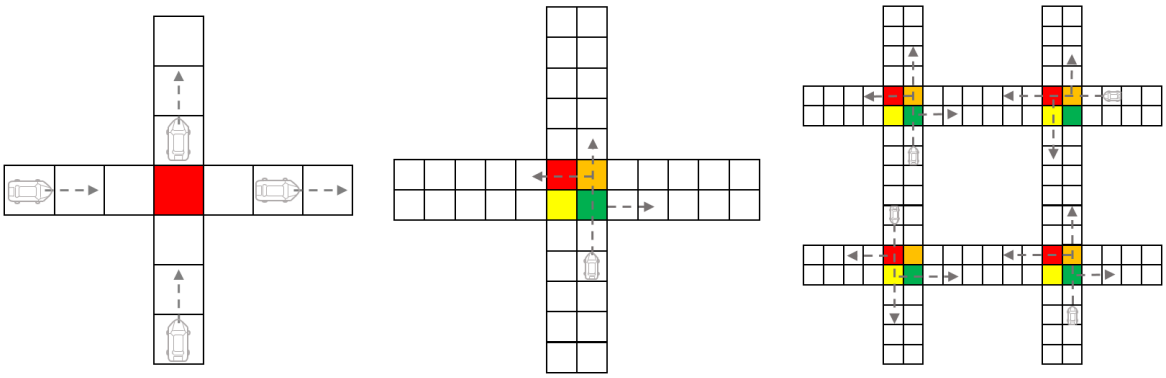


Figure 2: MJAs in each mode of traffic junction. Highlighted with colors.

neous agents, so that the solution of the problem is a shared policy which maximizes the cumulative discounted reward.

The problem is modeled using Dec-POMDP (Bernstein et al., 2002) using the tuple ($\mathcal{N}$, $\mathcal{S}$, $\{A_i\}$, $\{Z_i\}$, T, R, O). Where $\mathcal{N}$ is a finite set of agents, $\mathcal{S}$ is a set of states, $\{A_i\}$ is a set of actions for each agent i, $\{Z_i\}$ is a set of observations for each agent i, and T, R, O are the joint transition, reward, and observation models, respectively.

3.3 Environmental Setup

3.3.1 Micro Junction Agent

The intersection grid cells of the traffic junction environment are partitioned as shown in Figure 2. Each partitioned cells (four for each intersection except for the easy scenario) are homogeneous agents and control the vehicles in their governing area. These agents are called Micro Junction Agents (MJAs). In the easy mode, there is just one junction agent on the single path routes. For the medium and hard mode scenarios, there are four agents per junction. Because this is an autonomous environment, vehicles on a non-intersection grid cell go forward and stop if there is a vehicle in the forward cell. When a vehicle reaches the controlling area of MJA, the next time-step move of the vehicle is determined by the MJA. By adopting MJA, we change the subject of control policy and have the benefit of handling multiple vehicles.

3.3.2 Observations

The observation space of each MJA mainly consists of the features of the cars in its vision range. Our agent has a vision range of 5 cells (red bordered cells in Figure 3), i.e., the center of an MJA and its neighboring 4 cells. The number of features for each cell is 15, including the Boolean feature of whether crashes

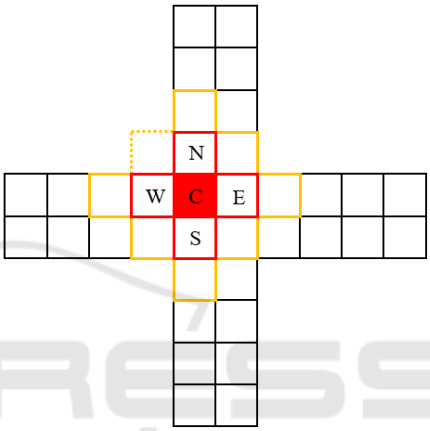


Figure 3: The observation of one junction agent.

have occurred at the cell and one-hot-encoded next cell position of the car in a cell, as shown in Table 1. These features are called positional features. In addition, the cell existence (i.e. whether a cell is in the environment) of the surrounding 13 cells are encoded (orange bordered cells in Figure 3).

The observation feature vector of an MJA is set as a concatenation of 5 positional features and the binary vector for cell existence. Therefore, the size of the observation vector is 15×5(each direction)$+13 = 88$.

3.3.3 Actions

The vehicle's actions are *gas*: go forward and *stay*: stop. However, when they are in the controlling areas of a MJA, they are governed and controlled by MJA's actions.

The action configuration of the MJA is to determine from which direction a vehicle can enter into its observation area. There are a total of five actions, one in each direction and none. If the action is 2 (East), only vehicles from the east can be accepted. Action 0 (None) means that no vehicle can enter. Because the

Table 1: Observation Features - Positional / Non-positional. Binary feature: (B), One-hot encoded feature:(O).

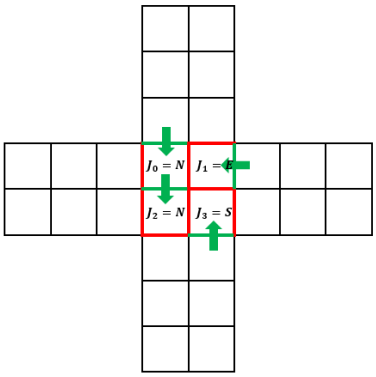
Features	Obs Type	Description
0 (B)	Pos	Existence of vehicle on each cell
1-13 (O)	Pos	Vehicle's next position
14 (B)	Pos	Crash occurrence
75-87 (O)	Non-Pos	Cell existence of the vehicle's next position

Table 2: The MJA's Action Configuration.

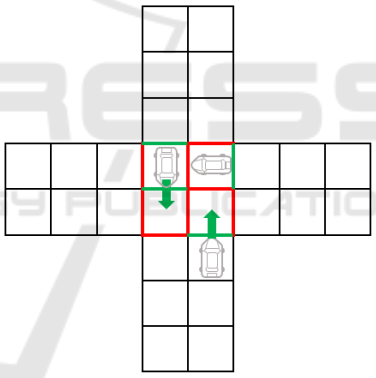
Index	Action
0	None
1	N
2	E
3	S
4	W

route of vehicles are known, MJAs can determine the direction of the vehicles. For example, as shown in Figure 4 (Junction J_3), if the vehicle comes in from the South and wants to move forward (North), only when the corresponding MJA's input action is South, the vehicle can go on to the next cell that is controlled by another MJA. This scheme remains the same even if there's more than one vehicle in the adjacent cell.

Vehicles that are already located on a MJA try to move forward, but if there is another MJA in that direction, they are controlled by both MJAs (Figure 4b Junction- J_0, J_1). A detailed mechanism of recurrent agents' action which is the rule of 'Agreement' is essential. Except for easy mode, all roads simulate a two-way street and each MJA is adjacent to two MJAs. One agent that possesses the vehicle on the cell makes it move forward and the other agent decides whether to accept it. If the relative out-direction of the vehicle and the absolute acceptance direction are the same, it will move.



(a) Junctions' action
$A = \{J_0 = N, J_1 = E, J_2 = N, J_3 = S\}$



(b) vehicles' movements

Figure 4: Junctions' action and vehicles' movements.

3.3.4 Reward

The vehicle agent's reward structure from baselines (Sukhbaatar et al., 2016) is used in this paper. There are collision penalty $r_{coll} = -10$, and time-step penalty $\tau r_{time} = -2\tau$ to discourage a traffic jam where τ is the number time steps passed since the vehicle appeared in the scenario. The reward for ith vehicle which is having C_i^t collisions and mean total reward at time t is:

$$r_i(t) = C_i^t r_{coll} + \tau_i r_{time} \quad (1)$$

$$G(t) = \frac{1}{N} \sum_{i=1}^{N} r_i(t) \quad (2)$$

where N is the total number of activated vehicles at time-step t. Averaging N vehicles' reward, the total mean reward G is used for all MJAs. Just by using the global reward in the training process, MJAs learned to coordinate the whole system without explicit cooperation.

3.4 Reinforcement Learning

To train MJAs, we construct a dueling double deep Q-learning neural network (DDDQN) (Wang et al., 2016) and use it for reinforcement learning. It is an enhanced architecture of DQN (Mnih et al., 2013) by adopting the target network (Double DQN) method, and separating state values and action advantages. In the experiments, we use an identical structure for both networks using 2 MLP layers (Pal and Mitra, 1992)

Table 3: Success rate comparison in the Traffic Junction.

Approach	Easy	Medium	Hard
CommNet	$93.0 \pm 4.2\%$	$54.3 \pm 14.2\%$	$50.2 \pm 3.5\%$
IC3Net	$93.0 \pm 3.7\%$	$89.3 \pm 2.5\%$	$72.4 \pm 9.6\%$
GA-Comm	99.7%	97.6%	82.3%
CENT-MLP	$97.7 \pm 0.9\%$	0%	0%
DICG-DE-MLP	$95.6 \pm 1.5\%$	$90.8 \pm 2.9\%$	$82.2 \pm 6.0\%$
TarMAC 2-round	$99.9 \pm 0.1\%$	-	$97.1 \pm 1.6\%$
MJA (Ours)	100.0%	$99.7 \pm 0.3\%$	$99.8 \pm 0.2\%$

Table 4: Comparison of success rate in harder mode.

	Harder 40-step	Harder 60-step
IQL with comm	94.3%	-
COMA	99.1%	-
CCOMA	99.3%	-
MJA (Ours)	$99.9\% \pm 0.1\%$	$99.9\% \pm 0.1\%$

Table 5: Traffic Junction environment configuration.

Difficulty	# MJA	# roads	Road dim.	N_{max}	P_{arrive}	Max steps
Easy	1	2	7×7	5	0.3	20
Medium	4	4	14×14	10	0.2	40
Hard	16	8	18×18	20	0.05	60
Harder	16	8	18×18	20	0.1	40 or 60

of 256 sizes. Observation scheme described in Sec 3.3.2 substitutes the need for observation embedding and complex architectures. We adopt independent Q-learning using parameter sharing (Foerster et al., 2016). Therefore, all agents are operated by the same parameterized policy.

4 RESULTS

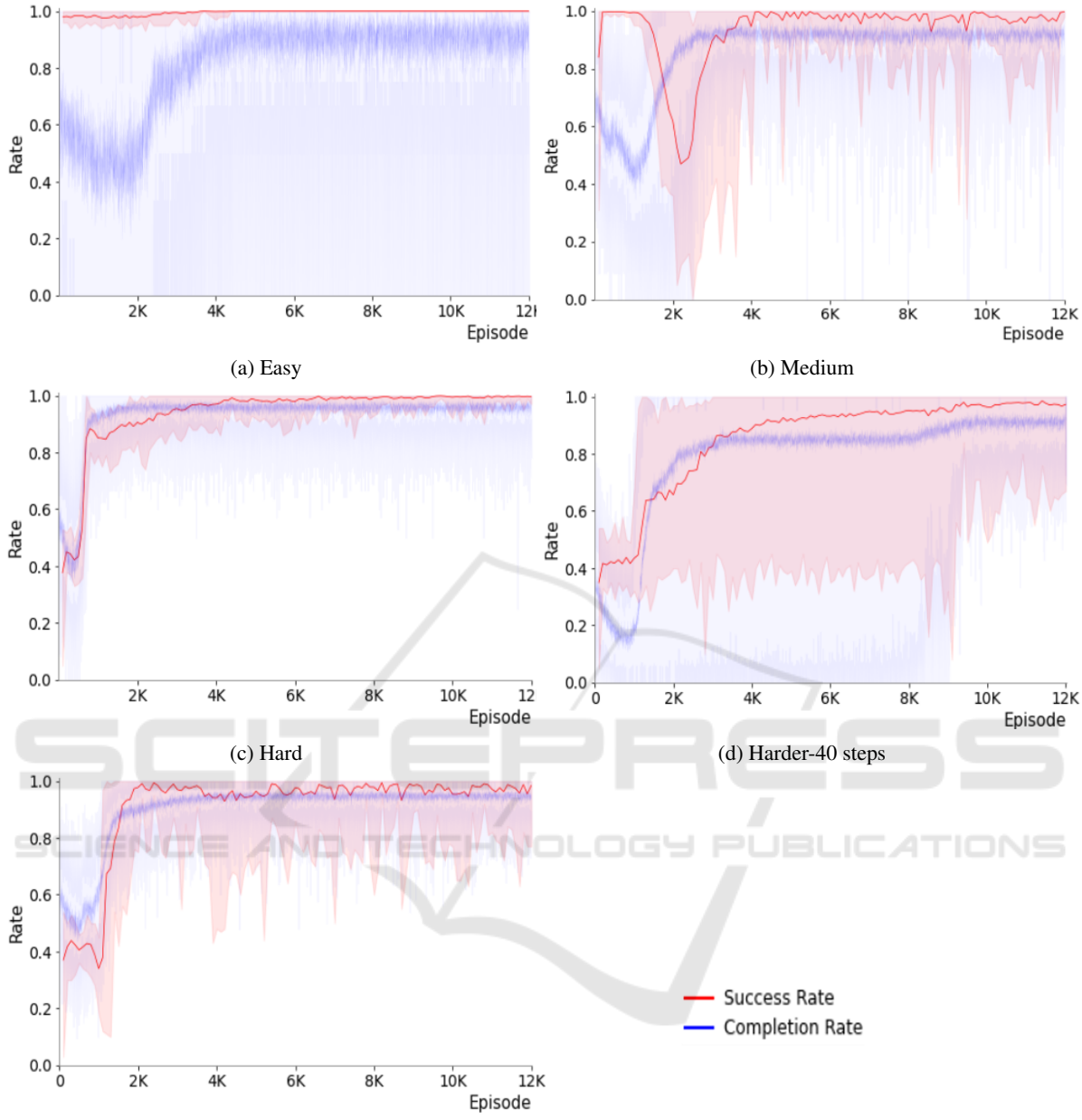
To evaluate and compare to other approaches, the table from (Li et al., 2020; Das et al., 2019) is used. Different from other research (Sukhbaatar et al., 2016; Singh et al., 2018; Liu et al., 2020; Su et al., 2020), our method does not adopt curriculum training (Bengio et al., 2009).

The configuration for four different difficulty modes is shown in Table 5. Note that a single MJA is used in Easy mode, which reduces the setup into a single-agent problem, although there are multiple moving vehicles.

The proposed model is trained for 12,000 episodes for each difficulty setup, where one episode is simulation of a scenario for the maximum number of time-steps determined by the max steps shown in Table 5. The definition of 'success rate' from (Li et al., 2020) is used, which is the ratio of episodes without any collision versus the number of episodes tested. The success rate in Figure 5 (red line) is calculated by running 100 evaluation episodes for every 100 training episodes. The best results compared to other methods are shown in Table 3. A harder mode is investigated, suggested by (Su et al., 2020), and compare to other methods that considered harder modes in their research. The result is in Figure 5 and Table 4.

The completion rate is plotted as the blue line in Figure 5 where, the completion rate refers to the ratio of vehicles that reached the destination versus the total vehicles that can arrive at the destination before the last time-step. in the same color. In the early stage on training, the policy tends to fall into a local optimum region where to achieve a high success rate, vehicles are not moved to avoid collision. Nevertheless, our learning scheme escapes from the local minimum and is able to find the optimal solution as shown in Figure 5a, 5b. The cause can be the balancing of two different rewards (collision penalty and time-step penalty which are described in Sec 3.3.4).

Comparing the results of the hard mode with the harder-60 steps it can be said that harder-60 converges faster to the optimal solution. This may be because, as P_{arrive} is increased, there is more opportunity of learning and thus, faster convergence. The reason for efficient learning may be because MJAs are homogeneous agents, parameter sharing is adopted, and independent Q-learning approach is used.



(a) Easy

(b) Medium

(c) Hard

(d) Harder-40 steps

(e) Harder-60 steps

Figure 5: Evaluations during training each difficulty mode. 20 random seeds are used for each difficulty. Red and blue lines indicate the average and the range of success rate and completion rate, respectively.

We also show that our method can perform stable training across 20 different random seeds. Furthermore, the hyperparameters used for difficulty modes are the same except for the learning rate ($lr = 0.5e-4$ for easy and $lr = 0.1e-4$ for the other modes).

5 CONCLUSIONS

In this paper, we introduce a novel multi-agent reinforcement learning (MARL) approach for traffic control. The proposed approach is tested on a traffic junction environment where multiple vehicles and junctions exist. A controller agent called Micro Junction Agent (MJA) is used for an autonomous intersection management (AIM) environment. Results show that

even without complex communication mechanisms, the traffic can be controlled and achieve high performance. Because the number of agents is not the function of vehicles but the function of intersections, it can be said that the proposed method is scalable. Future work may include applying our approach to more complex large-scale maps including more vehicles.

ACKNOWLEDGEMENTS

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2016R1D1A1B04933156)

REFERENCES

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. (2009). Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48.

Bernstein, D. S., Givan, R., Immerman, N., and Zilberstein, S. (2002). The complexity of decentralized control of markov decision processes. *Mathematics of operations research*, 27(4):819–840.

Buşoniu, L., Babuška, R., and De Schutter, B. (2010). Multi-agent reinforcement learning: An overview. *Innovations in multi-agent systems and applications-1*, pages 183–221.

Chouhan, A. P. and Banda, G. (2018). Autonomous intersection management: A heuristic approach. *IEEE Access*, 6:53287–53295.

Das, A., Gervet, T., Romoff, J., Batra, D., Parikh, D., Rabbat, M., and Pineau, J. (2019). Tarmac: Targeted multi-agent communication. In *International Conference on Machine Learning*, pages 1538–1546. PMLR.

Dresner, K. and Stone, P. (2008). A multiagent approach to autonomous intersection management. *Journal of artificial intelligence research*, 31:591–656.

Fiori, C., Arcidiacono, V., Fontaras, G., Makridis, M., Mattas, K., Marzano, V., Thiel, C., and Ciuffo, B. (2019). The effect of electrified mobility on the relationship between traffic conditions and energy consumption. *Transportation Research Part D: Transport and Environment*, 67:275–290.

Foerster, J. N., Assael, Y. M., De Freitas, N., and Whiteson, S. (2016). Learning to communicate with deep multi-agent reinforcement learning. *arXiv preprint arXiv:1605.06676*.

Gupta, J. K., Egorov, M., and Kochenderfer, M. (2017). Cooperative multi-agent control using deep reinforcement learning. In *International Conference on Autonomous Agents and Multiagent Systems*, pages 66–83. Springer.

Hartanti, D., Aziza, R. N., and Siswipraptini, P. C. (2019). Optimization of smart traffic lights to prevent traffic congestion using fuzzy logic. *TELKOMNIKA Telecommunication Computing Electronics and Control*, 17(1):320–327.

Hausknecht, M., Au, T.-C., and Stone, P. (2011). Autonomous intersection management: Multi-intersection optimization. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4581–4586. IEEE.

Khoza, E., Tu, C., and Owolawi, P. A. (2020). Decreasing traffic congestion in vanets using an improved hybrid ant colony optimization algorithm. *J. Commun.*, 15(9):676–686.

Lasley, P. (2019). 2019 urban mobility report.

Li, S., Gupta, J. K., Morales, P., Allen, R., and Kochenderfer, M. J. (2020). Deep implicit coordination graphs for multi-agent reinforcement learning. *arXiv preprint arXiv:2006.11438*.

Liu, Y., Wang, W., Hu, Y., Hao, J., Chen, X., and Gao, Y. (2020). Multi-agent game abstraction via graph attention neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7211–7218.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Pal, S. and Mitra, S. (1992). Multilayer perceptron, fuzzy sets, and classification. *IEEE Transactions on Neural Networks*, 3:683–697.

Parker, A. and Nitschke, G. (2017). How to best automate intersection management. In *2017 IEEE Congress on Evolutionary Computation (CEC)*, pages 1247–1254. IEEE.

Singh, A., Jain, T., and Sukhbaatar, S. (2018). Learning when to communicate at scale in multiagent cooperative and competitive tasks. *arXiv preprint arXiv:1812.09755*.

Su, J., Adams, S., and Beling, P. A. (2020). Counterfactual multi-agent reinforcement learning with graph convolution communication. *arXiv preprint arXiv:2004.00470*.

Sukhbaatar, S., Fergus, R., et al. (2016). Learning multiagent communication with backpropagation. *Advances in neural information processing systems*, 29:2244–2252.

Tan, M. (1993). Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the tenth international conference on machine learning*, pages 330–337.

Walraven, E., Spaan, M. T., and Bakker, B. (2016). Traffic flow optimization: A reinforcement learning approach. *Engineering Applications of Artificial Intelligence*, 52:203–212.

Wang, Z., Schaul, T., Hessel, M., Hasselt, H., Lanctot, M., and Freitas, N. (2016). Dueling network architectures for deep reinforcement learning. In *International conference on machine learning*, pages 1995–2003. PMLR.

Wei, H., Zheng, G., Gayah, V., and Li, Z. (2019). A survey on traffic signal control methods. *arXiv preprint arXiv:1904.08117*.