IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 2 June 2023, accepted 28 June 2023, date of publication 18 July 2023, date of current version 26 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3296537

RESEARCH ARTICLE

Flexible Traffic Signal Control via Multi-Objective Reinforcement Learning

TAKUMI SAIKI AND SACHIYO ARAI
Department of Urban Environment Systems, Division of Earth and Environmental Sciences, Graduate School of Science and Engineering, Chiba University, Chiba 263-8522, Japan

Corresponding author: Takumi Saiki (afma6906@chiba-u.jp)

This work was supported in part by the Industrial Technology Research Grant Program from the New Energy and Industrial Technology Development Organization (NEDO) of Japan under Grant P18010, and in part by the Japan Science and Technology Agency Support for Pioneering Research Initiated by the Next Generation (JST SPRING) under Grant JPMJSP2109.

ABSTRACT Deep reinforcement learning has been extensively studied for *traffic signal control* owing to its ability to process large amounts of information and achieving superior performance control. However, this method acquires flow-specific policies during learning. Thus, its performance under inexperienced traffic flows is not guaranteed. Moreover, the *traffic signal control* problem formulation assumes that the optimal policy differs for each traffic flow ratio owing to the trade-off between orthogonal roads at an intersection. Therefore, multiple policies must be switched to avoid performance decay for traffic flow changes. In this study, we use *multi-objective reinforcement learning* to determine the policy corresponding to each traffic flow ratio exhaustively. Subsequently, these policies are switched to the current traffic flow ratio to achieve flexible control over traffic flow changes. The proposed method achieves the shortest average travel times in all environments compared with rule-based and single-objective reinforcement learning methods for stationary traffic and traffic flows with varying flow ratios.

INDEX TERMS Traffic signal control, reinforcement learning, multi-objective optimization.

I. INTRODUCTION

Reinforcement learning (RL) methods have been widely studied in traffic signal control (TSC) as new alternatives to rule-based methods with a heavy hand computational burden. Generally, in rule-based methods, the goal is to reduce congestion and bias. However, guaranteeing optimal control with these methods is often challenging [1], [2].

By contrast, RL guarantees optimal convergence and is expected to obtain a policy (control law) with performance superior to rule-based methods. However, the computational power and required manual settings of the considered features have limited the early application of RL methods. The advent of deep reinforcement learning (DRL), which uses deep learning for feature extraction, has substantially improved performance owing to its ability to learn with more detailed information. Since then, several TSC methods based on DRL have been proposed [3], [4], [5], [6].

In this study, we focus on the following two features that render the application of RL to TSC challenging:

1) **Non-Markov properties**
   RL generally requires Markovian assumptions regarding the environment. Therefore, most current research addresses the control of stationary traffic flow. In general, actual traffic flows can also be assumed stationary for a particular period. However, several periods can have different flow rates. Thus, different Markov decision process models should represent these periods.
2) **Different policies are required for different traffic flows**
   The optimization target of RL is a scalar reward. This setup is natural for a problem with a single objective. However, in a multi-objective problem, providing multiple types of rewards is essential. Furthermore, if a trade-off exists between objectives, it may not be possible to obtain policies that consider the priority of each objective.
   Multi-objective reinforcement learning (MORL) is attracting attention in these fields. MORL is an

The associate editor coordinating the review of this manuscript and approving it for publication was Razi Iqbal.

TABLE 1. MDP/MOMDP element.

$\mathcal{S}$	state set
$\mathcal{A}$	the action set
$\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$	state transition probability function
$\boldsymbol{R} : \mathcal{S} \times \mathcal{R} \to \mathbb{R}$	Reward function
$\gamma \in [0, 1]$	discount rate

extension of RL applied to multi-objective optimization problems and can learn a Pareto policy that considers the priority of each objective. MORL is used in automated driving [7] and pedestrian simulation [8]. Moreover, the TSC problem presents a traffic flow trade-off between orthogonal roads. However, the time-varying nature of traffic flow implies that the priority of each road varies. Therefore, multiple policies tailored to different traffic flows are required to prevent performance degradation under changing traffic flows.

For (1), stable control has been achieved in [9], even under traffic conditions that are not experienced during training, whereas for (2), no studies have yet considered this trade-off.

In this paper, we propose a flexible TSC method and solve problem (2) via the following two steps: (i) We formulate the problem as a multi-objective optimization problem to minimize the travel time for each roadway. Subsequently, we use MORL to obtain Pareto policies and (ii) switch between the obtained policies according to the traffic flow ratio.

The remainder of this paper is organized as follows: Section II provides an overview of RL, Section III defines the research problem, Section IV describes the proposed method, Section V discusses the computer experiments, and Section VI summarizes the study.

II. BACKGROUND KNOWLEDGE

A. MULTI-OBJECTIVE REINFORCEMENT LEARNING

RL is a machine-learning method [10] that obtains a policy for an agent through trial and error in an environment that consists of a Markov decision process (MDP). MORL [11] extends RL to a multi-objective MDP (MOMDP). The MDP and MOMDP are defined as $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \boldsymbol{\mathcal{R}}, \gamma \rangle$. The elements of the MDP and MOMDP are presented in Table 1. Notably, the reward function $\boldsymbol{\mathcal{R}} \in \mathbb{R}^{\boldsymbol{n}}$ returns a reward vector for each objective. n indicates the number of objectives. The MDP can be understood as a special form of the MOMDP in an environment with a single objective ($n = 1$).

An agent's policy $\pi : \mathcal{S} \to \mathcal{A}$ represents the probability of selecting an action in each state. Equation (2) denotes the value of state s. It is computed from the rewards in state s and the expected value of future rewards.

The Q-value, which is the value of a pair of states and actions, is represented by Equation (3).

The optimal policy with respect to the optimal Q-value function $Q^*(s, a)$ is the action that maximizes $Q^*(s, a)$ for any given s, which is denoted by Equation (1). The agent's goal is to determine the optimal policy π^* satisfying Equation (1).

$$\pi^* = \arg\max_a Q^*(s, a) \tag{1}$$

$$\mathbf{V}^{\pi}(s) = \mathbb{E}_{\pi}\left\{\sum_{k=0}^{\infty} \gamma^k \boldsymbol{r}_{k+1}\right\} \tag{2}$$

$$\mathbf{Q}^{\pi}(s, a) = \mathbb{E}_{\pi}\left\{\sum_{k=0}^{\infty} \gamma^k \boldsymbol{r}_{k+1}\right\} \tag{3}$$

Note that the policy with weights (priorities) must be scalarized for each objective during the learning process to determine the optimal policy indicated in Equation (1) from the reward vector. The policy obtained in this process is known as the Pareto policy. For the value functions V^{π^i} and V^{π^j} that are obtained from the Pareto policies π^i and π^j in the MOMDP for objective $n \in N$, $V^{\pi_n^i} > V^{\pi_n^j}, \exists n \in N$ and $V^{\pi_n^i} \leq V^{\pi_n^j}, \forall n \in N$, π^i dominates π^j. A policy that is not dominated by all policies π is known as a Pareto optimal policy, and the set of Pareto optimal policies is referred to as the Pareto front.

1) SINGLE-POLICY AND MULTIPLE-POLICY APPROACH

The scalarization method from the reward vector can be divided into single-policy and multiple-policy approaches, as shown in Figure 1, depending on the scalarization procedure.

In the single-policy approach (Figure 1 top), the acquired reward is immediately scalarized using a scalarized function. The weight $\boldsymbol{w}$ is a predetermined priority for each objective. Moreover, the linear weighted sum presented in Equation (4) is used as a simple scalarization function, and the optimal policy presented in Equation (1) is obtained from the scalarized rewards using Equation (2) and Equation (3). The usual RL algorithm can be used in the single-policy approach; however, only one Pareto solution is obtained, and multiple training cycles with different weights are required to obtain a Pareto front.

$$R = \boldsymbol{w} \cdot \boldsymbol{r} = w_1 r_1 + w_2 r_2 + \ldots w_n r_n \tag{4}$$

$$w_1 + w_2 + \ldots w_n = 1 \tag{5}$$

By contrast, the multiple-policy approach (Figure 1 bottom) computes $\mathbf{V}$ and $\mathbf{Q}$ directly from $\boldsymbol{r}$ as vectors using Equation (2) and Equation (3). This approach scalarizes $\mathbf{V}$ and $\mathbf{Q}$ with weights $\boldsymbol{w}$ at the action-choosing time. No prior weight or the Pareto front in single training is required for this approach. However, a dedicated algorithm for learning as vectors is required.

B. DRL; APE-X DEEP Q-NETWORK

DRL which introduces deep learning for feature extraction, has significantly improved performance because this method can learn with more detailed information. In the DQN, the Q function is approximated by a deep neural network that enables learning in real-time even in continuous-valued and multi-dimensional state spaces [12]. A correlation exists

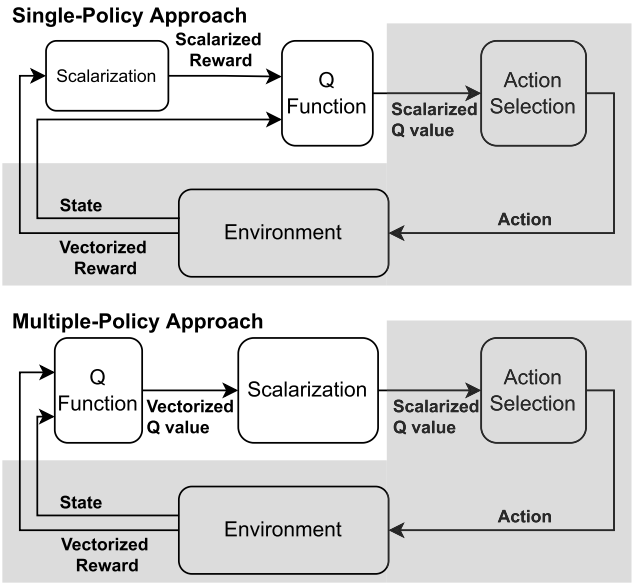

FIGURE 1. Single-policy approach(top) and multiple-policy approach (bottom) MORL workflow. Single-policy Q function inputs a state and outputs the scalarized Q-value. Multiple-policy Q function inputs the Q function and outputs the Q vector. Both methods share many parts (gray areas).

between the (s, a, s', r) tuples that the agents experience, and updating the network in the order in which they are experienced causes a bias. The DQN solves this problem by using experience replay, which randomly retrieves and updates the previous agents' experiences that are stored in the replay buffer.

However, experience replay requires large amounts of data to reduce sampling bias, and its simulation is time consuming. Ape-X deep Q-network (DQN) [13] is a deep RL method based on the DQN [12] accelerates learning by parallelizing the simulations.

C. MAX PRESSURE CONTROL

Max pressure control is a rule-based decentralized TSC method [2] with proven superior performance [14]. Conventional rule-based methods only observe the vehicles that enter an intersection. However, such methods do not consider congestion at adjacent intersections, making coordinated control between intersections impossible and complicating the entire traffic network optimization. By contrast, max pressure control adds the outbound traffic flow from the intersection to the observation range and sets control rules considering congestion at adjacent intersections.

III. PROBLEM DOMAIN AND RELATED RESEARCH

A. FLEXIBILITY TO CHANGES IN TRAFFIC FLOW

This study aims to achieve flexible control of various traffic flows in TSC using RL. A schematic of the intersection environment (isolated intersection) considered in this study is shown in Figure 6. The observation area consists of a controlling intersection and roads up to adjacent intersections. The traffic flow to be controlled is of vehicles at the intersection with traffic signals and on connecting roads. Two types of lanes connect at each intersection: one is to the intersection (inbound lane), and the other is from the intersection (outbound lane). Furthermore, the east-west and north-south roads are orthogonal to each intersection.

Changes in the traffic flow at the intersection can be classified into two categories: (1) changes in the traffic flow (the number of vehicles in the inbound section) and (2) changes in the traffic flow ratio (the ratio of the number of vehicles in the east-west and north-south directions). Cabrejas et al. [9] investigated the performance of RL-TSC for the problem (1). They reported that RL-TSC can achieve lower delays and fewer waiting vehicles compared to rule-based control, even when the traffic flow is higher than that experienced during the training phase.

Therefore, problem (2) can be solved if flexibility to changes in the traffic flow ratio can be achieved and realizing control that can be executed in a natural environment is feasible. However, as mentioned in the previous section, a trade-off exists between orthogonal roads, and no optimal policy is available to accommodate multiple traffic flows.

Therefore, we propose a new TSC method that obtains multiple policies for each traffic flow ratio and switches among them according to the traffic flow ratio.

B. RELATED RESEARCH

1) TSC via RL

Several previous studies have focused on acquiring a single objective optimal policy.

Genders and Razavi [3] used changes in the delay as rewards to minimize vehicle loss time. Gao et al. [15] used the change in the sum of the waiting times of all vehicles as a reward. Huo et al. [16] proposed a method to realize cooperative control among agents in multi-agent RL to control multiple intersections. In this study, the number of speed-restricted vehicles in the entire traffic network is used as a reward to achieve cooperative control.

All these indicators present an accurate image of the vehicle's condition. However, they are difficult to quantify in practice. Therefore, several methods have been proposed to use easily measurable indicators as rewards. Zheng et al. [17] investigated state representation and rewards in TSC using RL and demonstrated that sufficient performance can be obtained even with the queue length. Both Wei et al. [5] and Chacha et al. [6] used pressure, an idea from max pressure control, as a reward to achieve cooperative control when different agents control a multiple-intersection environment at each intersection.

2) TSC via MORL

The purpose of TSC is to facilitate traffic; however, multiple facilitation indicators are available. Therefore, several studies have introduced MORL to TSC and considered multiple objectives. Liao et al. [18] proposed a TSC method that considers traffic safety and maximum throughput.

Sometimes, some drivers ignore the red signal when their waiting time exceeds their impatience limit. For this, they introduced impatience as additional rewards with throughput. Gong et al. [19] proposed the MORL-based TSC method to obtain the policy that considers the trade-off between accident prevention and traffic jam reduction. This method focuses on the influence of vehicle location and signal switching on the probability of accidents. From the two rewards of safety and efficiency, TSC is implemented at the expense of efficiency in situations with an increasing probability of accidents. The method uses a single-policy approach to find a Pareto solution. As safety is an objective and a parameter that does not change over time, only one type of policy is required. However, multiple policies are required if the objectives' weights vary over time, rendering the single-policy method unsuitable.

3) TSC THAT SWITCHES MULTIPLE POLICIES

Focusing on the priority of each road, Abdoos et al. [20] proposed a method that switches between a control considering the wide-area traffic network and that considering only the vicinity of the intersection. This method trains two policies. One policy uses rewards from all roads adjacent to the intersection. The other policy uses rewards related to arterial roads only. Subsequently, the method switches between the two policies based on rules, depending on the traffic situation.

IV. PROPOSED APPROACH

This study focuses on MORL for acquiring multiple control methods according to traffic flow ratios. Owing to trade-offs between orthogonal intersections, obtaining a Pareto policy is necessary to implement optimal control over the traffic flow. However, a single Pareto policy is not always optimal owing to changing traffic flows. Therefore, it is necessary to obtain multiple Pareto policies with different weights exhaustively. Notably, two methods are required to select the optimal Pareto policy for the current traffic flow. In the MORL framework, the weights of each road replace the traffic flow ratio, and rewards for each objective replace the rewards for each orthogonal road. Our proposed method considers the following two-step approach:

1) Acquiring multiple Pareto policies using MORL. MORL learns the $\mathbf{Q}$ vector and $\mathbf{V}$ vector from $\boldsymbol{r}$.
2) Calculating appropriate weights for selecting Pareto policy using the current traffic condition. With weights and $\mathbf{Q}$, $\mathbf{V}$, we can choose a single policy to control the traffic signal.

A. TERMINOLOGY

The terminology for the proposed method is defined as follows.

Phase: *Phase* is a vector that indicates the permitted direction of traffic at an intersection.

Policy: In this study, we define a policy as a traffic signal switching law for a given traffic flow ratio.

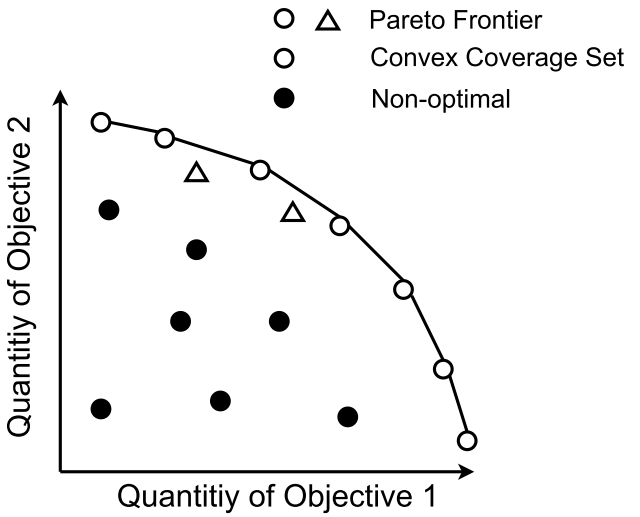

FIGURE 2. Pareto frontier and CCS.

Reward Vector: The reward in this study is the optimization goal in RL. Because MORL is used in this study, the rewards are the measured vectors for each orthogonal road.

State Vector: Based on the proposal by Wei et al. [5] and Chacha et al. [6], this study uses the number of vehicles in each lane as an input to the RL agent.

B. OBTAINING MULTIPLE POLICIES

1) ENVELOPE MOQ-LEARNING

The acquisition of multiple Pareto policies requires low computational complexity and exhaustiveness of Pareto policies. Therefore, we employed envelope MOQ-learning [21] to fulfill this requirement. Envelope MOQ-learning is an MORL method for obtaining a convex coverage set (CCS) in one training. CCS is the convex hull of the Pareto front in the Pareto solution set, as illustrated in Figure 2. Although the Pareto front includes a nonconvex solution set, only the CCS is obtained because envelope MOQ-learning scalarizes the weights per objective using linear summation.

The algorithm for envelope MOQ-learning is presented in Algorithm 1. The goal of the envelope MOQ-learning is to estimate the Q vector before scalarization. The original loss function L^A for this objective is presented in Equation (6).

$$L^A(\theta) = \mathbb{E}_{s,a,\boldsymbol{w}}\left[||\boldsymbol{y} - \boldsymbol{Q}(s, a, \boldsymbol{w}; \theta)||_2^2\right] \tag{6}$$

As the Pareto front contains numerous discrete solutions, the optimal loss plane is not flat; thus, directly optimizing the loss function L^A is challenging. Therefore, envelope MOQ-learning introduces an auxiliary loss function L^B, as Equation (7) indicates. The optimization function is applied to the weighted sum of the two loss functions using Equation (8), and convergence to the optimal solution is achieved by gradually increasing the weight of L^A. This method is known as homotopy optimization.

$$L^B(\theta) = \mathbb{E}_{s,a,\boldsymbol{w}}\left[|\boldsymbol{w}^{\mathrm{T}}\boldsymbol{y} - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{Q}(s, a, \boldsymbol{w}; \theta)|\right] \tag{7}$$

$$\nabla_\theta L(\theta) = (1 - \lambda)\nabla_\theta L^A(\theta) + \lambda\nabla_\theta L^B(\theta) \tag{8}$$

Algorithm 1 Envelope MOQ-Learning

Require: weight distribution D_w, trajectory p_λ, balance weight λ (increasing from 0 to 1)

replay $\mathcal{D}_\tau$, initialize network $\boldsymbol{Q}_\theta$, $\lambda = 0$.
for *episode* $= 1, \ldots, M$ **do**
  sample the linear weights $w \sim \mathcal{D}_w$.
  for $t = 0, \ldots, N$ **do**
    observe s_t and ϵ-greedy action selection

$$a_t = \begin{cases} \text{random action } a \in A & (\epsilon) \\ \max_{a \in A} w^{\mathrm{T}} \boldsymbol{Q}(s_t, a_t, r_t, s_{t+1}) & (1-\epsilon) \end{cases}$$

    Vector reward r_t, observe next state s_{t+1}. Store transition (s_t, a_t, r_t, s_{t+1}) in $\mathcal{D}_\tau$.
    if Update **then**
      Sample N_τ transitions $(s_t, a_t, r_t, s_{t+1}) \sim \mathcal{D}_\tau$ and Sample N_w weights $W = \{w_i \sim \mathcal{D}_w\}$
      for all $1 \leq i \leq N_w$ and $1 \leq j \leq N_\tau$ **do**
        Compute $y_{ij} = (\mathcal{T}\boldsymbol{Q})_{ij} =$

$$\begin{cases} \boldsymbol{r}_j, & s_{j+1} \text{ is in terminal states} \\ \boldsymbol{r}_j + \gamma \arg_Q \max_{\substack{a \in \mathcal{A} \\ \boldsymbol{w}' \in W}} \boldsymbol{w}_i^{\mathrm{T}} \boldsymbol{Q}\left(s_{j+1}, a, \boldsymbol{w}'; \theta\right), & \text{(otherwise)} \end{cases}$$

      end for
      Q_θ updated using Equation (8).
      λ increased for each P_λ trajectories
    end if
  end for
end for

TABLE 2. Notation list.

Notation	Meaning
$I \in \mathcal{I}$	Intersection set in environment
$l \in L$	Lane sets in the entire environment
$L(I)$	Lane set connected to intersection I
L_{EW}	East-west direction lane set
L_{NS}	North-south direction lane set
L^{OUT}	Outgoing lane set
L^{IN}	Incoming lane set
F_{EW}	East-west vehicle in flow [veh/s]
F_{NS}	North-south vehicle in flow [veh/s]
$p \in P$	Set of vehicles in lanes
t_g	Time of the green phase [s]
t_y	Time of the yellow phase [s]
t_r	Time of the red phase [s]
$v_t(l)$	Number of vehicles in lane l at time t
$q_t(l)$	Queue length of lane l at time t

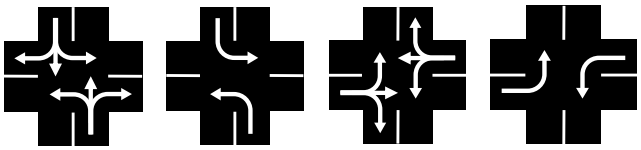

FIGURE 3. Action Set.

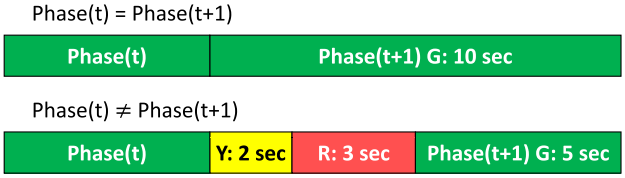

FIGURE 4. Phase Changing setting.

The multiple-policy approach requires more samples for training than the single-policy method because it is necessary to determine a policy for each weight. In envelope MOQ-learning, the state transition probabilities are independent of the weights w. Envelope MOQ-learning improves the sample efficiency by updating the network using experience acquired by other weights regarding hindsight experience replay [22].

2) STATE REPRESENTATION

The notation for the environment is listed in Table 2.

The agent controlling intersection I observes the number of vehicles $\{v(l), l \in I(L)\}$ in each lane connected to intersection I and the current signal indication $p_t(I)$ as a one-hot vector at time t. Therefore, the observed dimensions of the agent are $|L(I)| + 4$.

3) ACTION SETTING

The agent action involves selecting the present phase p_{t+1} in the next step and selecting one among four types of phases, including the left-turn-only phases shown in Figure 3. The phase change following this action selection is depicted in Figure 4. If the selected phase differs from the previous phase, this phase is changed to a new phase after the yellow and red phases. In this study, when the same indication is continued, the green phase is shown for 10 s, and when it changes, the yellow and red phases of $t_y = 2$ s $t_r = 3$ s, respectively, are interspersed, followed by a green signal of 5 s.

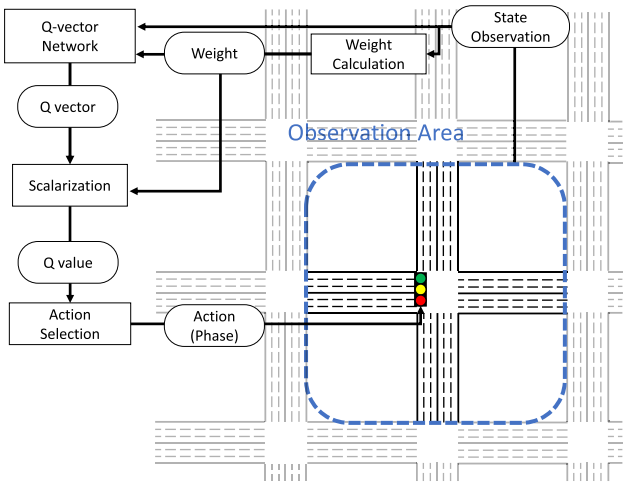

FIGURE 5. Proposed method of control flow.

4) REWARD FUNCTION

The concept of max pressure control has been applied to the state and reward representation in recently proposed RL methods [5], [6]. Our method also implements this concept as the state and reward representation. Here, pressure is expressed in $q(l)$ using the queue lengths $l \in L_{\mathrm{IN}}$ of the incoming lane and $l \in L_{\mathrm{OUT}}$ of the outgoing lane for the roads connected to the intersection. The pressure is an index that increases along a direction that does not worsen the congestion occurring at adjacent intersections, and it is suitable for rewards in multi-agent control when intersections are required to cooperate. Moreover, the pressure is used as a vector reward for each orthogonal road, as Equation (10) indicates.

$$P_t(L(I)) = -\left(\sum_{L(I)^{\mathrm{IN}}} q_t(l) - \sum_{L(I)^{\mathrm{OUT}}} q_t(l)\right) \tag{9}$$

$$\boldsymbol{r}_t = (P_t(L(I)_{NS}), P_t(L(I)_{EW})) \tag{10}$$

5) PARAMETER SHARING

The learning process may be more complicated in a multi-agent environment than in a single-agent environment because the actions of each agent affect the state transitions of the environment. We use parameter sharing, where the policies are shared among agents, to accelerate the learning process and improve performance by allowing agents to acquire various experiences.

6) PARALLELIZATION

Envelope MOQ-learning differs from the DQN only in the network structure [12], whereas the primary learning structure is the same. Therefore, we parallelized the actor based on Ape-X DQN [13] to accelerate the experiments.

C. SWITCHING POLICIES

We use envelope MOQ-learning to acquire policies for multiple road priorities. Subsequently, we switch policies according to the procedure summarized in Figure 5. The acquired neural network requires two weight inputs to compute the action: (1) at the input and (2) as a scalarization after the output.

Algorithm 2 Weight Caluculation

Require: East-west incoming lane $L(I)_{EW}$, north-south incoming lane $L(I)_{NS}$, number of vehicles per lane $L(I)$.

Ensure: weight vector $\boldsymbol{w}$

1: $w_{NS} = \dfrac{\sum_{L(I)_{NS}} v(l)}{\sum_{L(I)_{EW}} v(l) + \sum_{L(I)_{NS}} v(l)}$

2: $w_{EW} = \dfrac{\sum_{L(I)_{EW}} v(l)}{\sum_{L(I)_{EW}} v(l) + \sum_{L(I)_{NS}} v(l)}$

3: $\boldsymbol{w} = (w_{EW}, w_{NS})$

In the TSC problem, the weights can be viewed as priorities per road; that is, the traffic flow ratios per road. The exact traffic flow ratio can be determined by considering the entire intersection environment. In this study, a simple method was used to determine the priority based on the ratio of the number of vehicles using the algorithm presented in Algorithm 2.

V. COMPUTER EXPERIMENTS

The performance of the proposed method was evaluated by conducting computer experiments using the traffic simulator SUMO [23]. We used a machine with Corei-9 10980XE, RTX-2080 Ti with 128 GB RAM. The performance of the proposed method was compared with that of the rule-based and single-objective RL methods.

Two experimental environments were established: an isolated intersection environment, and an environment with four intersections arranged in a grid, as illustrated in Figure 6. The distances between intersections and from the edges of the environments to the intersection were 200 m in both environments.

A. COMPARISON METHODS

The rule-based max pressure control and single-objective RL methods were used for comparison. The single-objective RL method had the same state representation and actions as the proposed method, and its reward was immediately scalarized.

B. TRAFFIC FLOW

We used a 60 min simulation per episode with a stationary traffic flow of $F_{\mathrm{EW}} = F_{\mathrm{NS}} = 0.3$ veh/sec for training. During the test, two types of traffic flows were used: (1) the same stationary traffic flow as in the training simulation, and (2) a traffic flow where the traffic flow ratio changed every 10 min while maintaining $F_{\mathrm{EW}} + F_{\mathrm{NS}} = 0.6$ veh/sec according to Table 3. The simulation was performed for 80 min in a traffic flow environment. The environment considered 2×2 intersections and the traffic flow on two roads in the same direction was assumed to vary identically.

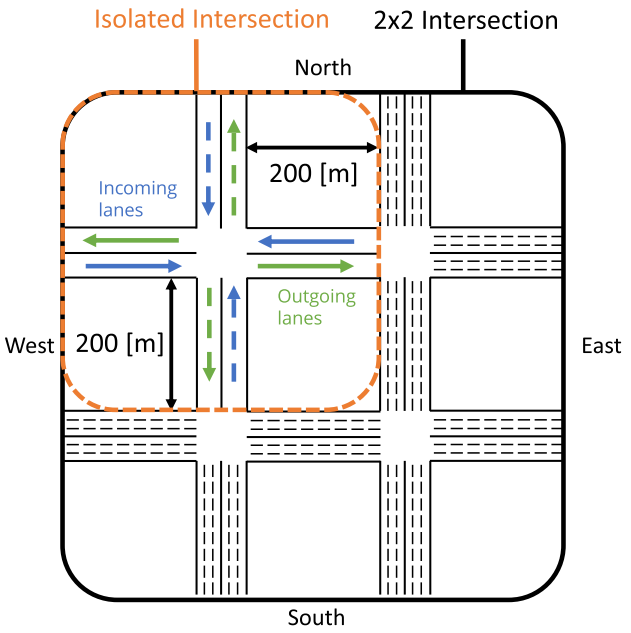

FIGURE 6. Experimental environments (Isolated intersection and 2 × 2 Intersections).

TABLE 3. Traffic flow setting in ratio-diverse traffic flow (80 [min]).

time[min]	0-	10-	20-	30-	40-	50-	60-	70-
F_{EW}	4	3	2	1	1	2	3	4
F_{NS}	1	2	3	4	4	3	2	1

C. EVALUATION

The moving average of the vehicles that appeared in the environment during each 10 min period was used as a comparison index. The mean and standard deviation of the moving average for each of the ten tests accounted for performance differences owing to random numbers.

D. EXPERIMENTAL RESULTS

1) ISOLATED INTERSECTION ENVIRONMENT

We compared the performance of the three methods for the isolated intersection, as illustrated in Figure 6. The average travel times of the three methods under the same training traffic flow are summarized in Figure 7, and the average and standard deviation of the ten trials of the average travel time every 10 min are listed in Table 4. The proposed method reduced the average travel time compared with the single-objective RL method. Moreover, the average travel time was comparable to the rule-based max pressure control. The performance of the single-objective RL method was poor than that of the proposed method under the learned traffic flow owing to the policy change during congestion. Even if the traffic flow is stationary at intersections, the priority of each road constantly changes owing to congestion.

The experiments considering changing traffic flow ratios are illustrated in Figure 8 and Table 5. The performance of the single-objective RL method was poor and less flexible because it could not follow the changes in the traffic flow

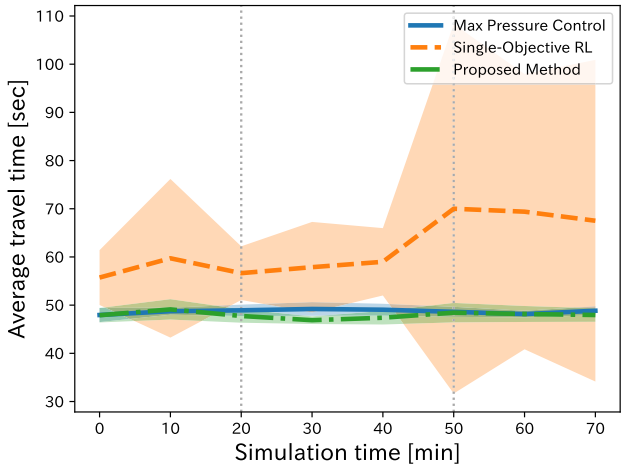

FIGURE 7. Average travel time of (1) Max Pressure Control, (2) Single-Objective RL, and (3) Proposed Method under stable traffic flow.

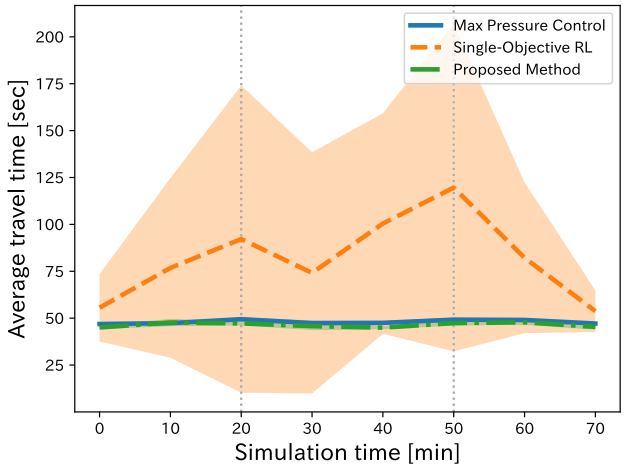

FIGURE 8. Average travel time of (1) Max Pressure Control (2) Single-Objective RL (3) Proposed Method under radio-diverse traffic flow.

ratios. The proposed and max pressure control methods maintained stable average travel times in both experiments. The max pressure control and proposed methods could follow the changes in the road priority by prioritizing congested roads and switching the Pareto policies, respectively. Thus, the proposed method is flexible for both stationary and dynamic traffic flows.

2) MULTIPLE INTERSECTION ENVIRONMENT

We compared the performance of the three methods in multi-agent control of a multiple-intersection environment. In particular, the average travel times of the three methods under a stationary traffic flow are summarized in Figure 9, and the average and standard deviation of the ten trials for the average travel time every 10 min are listed in Table 6.

The proposed method consistently achieved a short average travel time. This indicates that the proposed method can obtain stable Pareto policies even in a multi-agent environment. Unlike the experiment at the isolated

TABLE 4. Average travel time of (1) Max Pressure Control, (2) Single-Objective RL, and (3) Proposed Method under stable traffic flow in an isolated intersection environment.

Time [min]	[0,10)	[10, 20)	[20,30)	[30,40)	[40,50)	[50,60)	[60,70)	[70,80)
Max Pressure	48.0 ± 1.5	$\mathbf{48.7 \pm 0.7}$	48.9 ± 1.2	49.2 ± 1.4	49.1 ± 1.2	$\mathbf{48.6 \pm 1.0}$	$\mathbf{48.1 \pm 0.7}$	48.8 ± 0.9
Single-Policy	55.7 ± 5.69	59.7 ± 16.5	56.6 ± 5.57	57.9 ± 9.39	59.0 ± 6.99	70.0 ± 38.3	69.4 ± 28.6	67.5 ± 33.4
Proposed	$\mathbf{47.9 \pm 1.49}$	49.1 ± 2.11	$\mathbf{47.7 \pm 1.39}$	$\mathbf{46.9 \pm 0.8}$	$\mathbf{47.4 \pm 1.4}$	48.4 ± 2.0	48.2 ± 1.7	$\mathbf{48.0 \pm 1.4}$

TABLE 5. Average travel time of (1) Max Pressure Control, (2) Single-Objective RL, and (3) Proposed Method under ratio-diverse traffic flow in an isolated intersection environment.

Time [min]	[0,10)	[10, 20)	[20,30)	[30,40)	[40,50)	[50,60)	[60,70)	[70,80)
Max Pressure	46.9 ± 0.7	$\mathbf{47.3 \pm 1.2}$	49.5 ± 1.01	47.4 ± 1.5	47.5 ± 1.2	49.2 ± 1.1	49.1 ± 0.7	47.1 ± 0.8
Single-Policy	55.6 ± 18.0	76.8 ± 47.9	92.2 ± 81.9	74.2 ± 64.3	100.4 ± 58.8	120.0 ± 87.3	82.2 ± 40.2	53.7 ± 11.1
Proposed	$\mathbf{45.0 \pm 1.0}$	47.7 ± 2.0	$\mathbf{47.2 \pm 1.2}$	$\mathbf{45.8 \pm 2.2}$	$\mathbf{44.9 \pm 1.0}$	$\mathbf{47.4 \pm 1.4}$	$\mathbf{47.8 \pm 1.5}$	$\mathbf{45.4 \pm 1.6}$

TABLE 6. Average travel time of (1) Max Pressure Control, (2) Single-Objective RL, and (3) Proposed Method under stationary traffic flow at multiple intersections. The proposed method achieves the same performance as the existing methods for the learned traffic flows.

Time [min]	[0,10)	[10, 20)	[20,30)	[30,40)	[40,50)	[50,60)	[60,70)	[70,80)
Max Pressure	99.2 ± 3.2	107.5 ± 7.2	111.8 ± 12.6	112.0 ± 6.9	111.4 ± 7.7	109.6 ± 6.4	111.7 ± 5.9	102.9 ± 4.7
Single-Policy	75.8 ± 0.89	75.6 ± 1.20	75.5 ± 1.6	75.4 ± 2.0	75.3 ± 1.1	74.9 ± 1.8	76.3 ± 1.3	74.4 ± 1.5
Proposed	$\mathbf{73.8 \pm 1.5}$	$\mathbf{74.6 \pm 1.5}$	$\mathbf{73.9 \pm 0.9}$	$\mathbf{74.5 \pm 1.3}$	$\mathbf{74.9 \pm 1.2}$	$\mathbf{74.5 \pm 1.1}$	$\mathbf{74.8 \pm 1.1}$	$\mathbf{73.2 \pm 1.5}$

TABLE 7. Average travel time of (1) Max Pressure Control (2) Single-Objective RL, and (3) Proposed Method under ratio-diverse traffic flow at multiple intersections. Only the proposed method achieved stable average travel times for traffic flow changes not experienced during training.

Time [min]	[0,10)	[10, 20)	[20,30)	[30,40)	[40,50)	[50,60)	[60,70)	[70,80)
Max Pressure	116.3 ± 11.8	149.2 ± 15.3	127.2 ± 11.0	164.6 ± 17.9	221.0 ± 13.7	216.1 ± 11.5	170.9 ± 26.3	130.1 ± 10.3
Single-Policy	76.9 ± 2.5	76.1 ± 1.7	76.6 ± 1.2	82.0 ± 4.6	87.0 ± 22.6	82.2 ± 19.1	76.3 ± 2.2	79.5 ± 8.2
Proposed	$\mathbf{76.0 \pm 1.5}$	$\mathbf{75.1 \pm 0.7}$	$\mathbf{75.1 \pm 0.6}$	$\mathbf{75.2 \pm 1.8}$	$\mathbf{74.3 \pm 1.6}$	$\mathbf{73.9 \pm 1.0}$	$\mathbf{74.1 \pm 0.8}$	$\mathbf{73.6 \pm 0.9}$

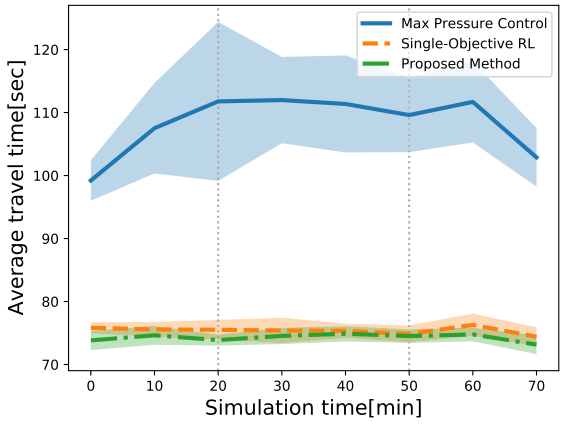

FIGURE 9. Average travel time of (1) Max Pressure Control, (2) Single-Objective RL, and (3) Proposed Method under stable traffic flow in 2 × 2 intersections environment.

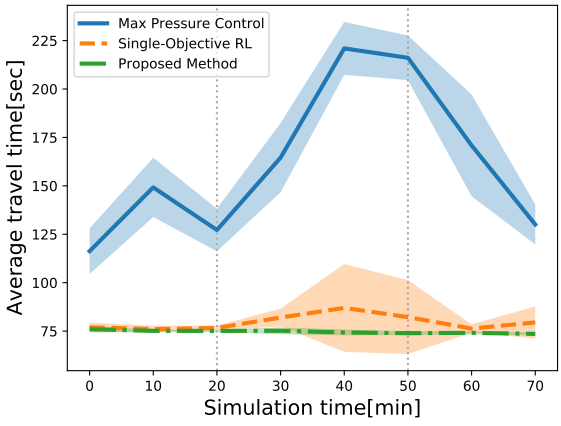

FIGURE 10. Average travel time of (1) Max Pressure Control, (2) Single-Objective RL, and (3) Proposed Method under ratio-diverse traffic flow at 2 × 2 intersection environment.

intersection, the average travel time of the max pressure control method was significantly worse than that of the other two methods, and the average travel times of the proposed and single-objective RL methods were almost the same. The superior performance of the single-objective method compared with that of the isolated intersections can be attributed to the generalization of the learning policy through parameter sharing.

The experiments considering changing traffic flow ratios are illustrated in Figure 10 and Table 7. Although the average travel time of the single-objective RL method deteriorated during the experiment, the proposed method always maintained the same average travel time. In a multi-agent environment, the traffic flow changes not only because of the incoming traffic flow from outside but also owing to the actions of other agents. Experimental results indicate that only the proposed method can suppress the performance deterioration caused by these two changes. This result supports the hypothesis that the proposed method is flexible to traffic flow ratio changes.

VI. CONCLUSION

A. SUMMARY

In this paper, we proposed an RL-based TSC considering the temporal variability of traffic flow and the inherent trade-offs between adjacent roads.

Although RL has been applied in various fields in recent years, the optimization goal must be expressed as a reward for scalar values. Therefore, MORL is used in multi-objective environments where trade-offs need to be considered.

This study focuses on the changing tradeoffs between orthogonal roads for the TSC problem. At intersections, it is necessary to consider the tradeoffs among roads and obtain a Pareto policy according to the traffic flow at the time of control. However, a single Pareto policy may result in poor performance because of the time-varying traffic flow. Therefore, to achieve flexible control according to the traffic flow, we proposed a method that employs a multi-policy approach and selects Pareto policies based on observed conditions post-training. The proposed method can be introduced in other fields because it is widely applicable to problems wherein the priority of each objective changes over time.

B. FUTURE RESEARCH

A possible future development is the introduction of hierarchical control. In the proposed method, the policy is switched by considering only the area around the intersection to be controlled. However, more global decisions can be made by considering traffic conditions at adjacent intersections. Hierarchical control, which separates actual signal switching from road priority decisions, is therefore considered suitable because Pareto policy learning does not require global decisions.

REFERENCES

[1] C. Gershenson, "Self-organizing traffic lights," 2004, *arXiv:nlin/0411066*.
[2] P. Varaiya, "Max pressure control of a network of signalized intersections," *Transp. Res. C, Emerg. Technol.*, vol. 36, pp. 177–195, Nov. 2013.
[3] W. Genders and S. Razavi, "Using a deep reinforcement learning agent for traffic signal control," 2016, *arXiv:1611.01142*.
[4] H. Wei, G. Zheng, H. Yao, and Z. Li, "IntelliLight: A reinforcement learning approach for intelligent traffic light control," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 2496–2505.
[5] H. Wei, C. Chen, G. Zheng, K. Wu, V. Gayah, K. Xu, and Z. Li, "PressLight: Learning max pressure control to coordinate traffic signals in arterial network," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 1290–1298.
[6] H. W. C. Chen, N. Xu, G. Zheng, M. Yang, Y. Xiong, K. Xu, and Z. Li, "Toward a thousand lights: Decentralized deep reinforcement learning for large-scale traffic signal control," in *Proc. 34th AAAI Conf. Artif. Intell. (AAAI)*, New York, NY, USA, 2020, pp. 1–8.
[7] M. Kim, S. Lee, J. Lim, J. Choi, and S. G. Kang, "Unexpected collision avoidance driving strategy using deep reinforcement learning," *IEEE Access*, vol. 8, pp. 17243–17252, 2020.
[8] N. B. Ravichandran, F. Yang, C. Peters, A. Lansner, and P. Herman, "Pedestrian simulation as multi-objective reinforcement learning," in *Proc. 18th Int. Conf. Intell. Virtual Agents*, Nov. 2018, pp. 307–312.
[9] A. Cabrejas-Egea, R. Zhang, and N. Walton, "Reinforcement learning for traffic signal control: Comparison with commercial systems," *Transp. Res. Proc.*, vol. 58, pp. 638–645, 2021.
[10] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.
[11] D. M. Roijers, P. Vamplew, S. Whiteson, and R. Dazeley, "A survey of multi-objective sequential decision-making," *J. Artif. Intell. Res.*, vol. 48, pp. 67–113, Oct. 2013.
[12] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing Atari with deep reinforcement learning," 2013, *arXiv:1312.5602*.
[13] D. Horgan, J. Quan, D. Budden, G. Barth-Maron, M. Hessel, H. van Hasselt, and D. Silver, "Distributed prioritized experience replay," 2018, *arXiv:1803.00933*.
[14] W. Genders and S. Razavi, "An open-source framework for adaptive traffic signal control," 2019, *arXiv:1909.00395*.
[15] J. Gao, Y. Shen, J. Liu, M. Ito, and N. Shiratori, "Adaptive traffic signal control: Deep reinforcement learning algorithm with experience replay and target network," 2017, *arXiv:1705.02755*.
[16] Y. Huo, Q. Tao, and J. Hu, "Cooperative control for multi-intersection traffic signal based on deep reinforcement learning and imitation learning," *IEEE Access*, vol. 8, pp. 199573–199585, 2020.
[17] G. Zheng, X. Zang, N. Xu, H. Wei, Z. Yu, V. Gayah, K. Xu, and Z. Li, "Diagnosing reinforcement learning for traffic signal control," 2019, *arXiv:1905.04716*.
[18] L. Liao, J. Liu, X. Wu, F. Zou, J. Pan, Q. Sun, S. E. Li, and M. Zhang, "Time difference penalized traffic signal timing by LSTM Q-network to balance safety and capacity at intersections," *IEEE Access*, vol. 8, pp. 80086–80096, 2020.
[19] Y. Gong, M. Abdel-Aty, J. Yuan, and Q. Cai, "Multi-objective reinforcement learning approach for improving safety at intersections with adaptive traffic signal control," *Accident Anal. Prevention*, vol. 144, Sep. 2020, Art. no. 105655.
[20] M. Abdoos and A. L. Bazzan, "Hierarchical traffic signal optimization using reinforcement learning and traffic prediction with long-short term memory," *Exp. Syst. Appl.*, vol. 171, Jun. 2021, Art. no. 114580.
[21] R. Yang, X. Sun, and K. Narasimhan, "A generalized algorithm for multi-objective reinforcement learning and policy adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 14636–14647.
[22] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, O. P. Abbeel, and W. Zaremba, "Hindsight experience replay," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5048–5058.
[23] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wiessner, "Microscopic traffic simulation using SUMO," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 2575–2582.

TAKUMI SAIKI received the B.S. and M.S. degrees in artificial intelligence from Chiba University, in 2022, where he is currently pursuing the Ph.D. degree. His research interests include intelligent transportation systems and multi-objective optimization.

SACHIYO ARAI received the B.S. degree in control engineering from Keio University and the M.S. and Ph.D. degrees in mathematical statistics and artificial intelligence from the Tokyo Institute of Technology, in 1998. She was with Sony Corporation for two years after receiving the B.S. degree. After receiving the Ph.D. degree, she spent a year as a Research Associate with the Tokyo Institute of Technology. She was a Postdoctoral Fellow with The Robotics Institute, Carnegie Mellon University, from 1999 to 2001, and a Visiting Associate Professor with the Department of Social Informatics, Kyoto University, from 2001 to 2004. She is currently a Professor with the Graduate School of Engineering, Chiba University. She is also working on industry-academia collaboration with several major companies; automotive, steel, and construction. She has authored several articles and two technical books related to reinforcement learning. Her research interests include machine learning and control theory. She is also a Board Member of the Japanese Society for Artificial Intelligence, serves on the program committees for several prominent international conferences, and has two competitive research funds in Japan.