Received 1 June 2023, accepted 19 July 2023, date of publication 26 July 2023, date of current version 2 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3298953

RESEARCH ARTICLE

A Deep Reinforcement Learning-Based Two-Dimensional Resource Allocation Technique for V2I Communications

HEETAE JIN, JEONGBIN SEO, JEONGHUN PARK, AND SUK CHAN KIM, (Senior Member, IEEE)
Department of Electronics Engineering, Pusan National University, Busan 609-735, South Korea

Corresponding author: Suk Chan Kim (sckim@pusan.ac.kr)

This work was supported in part by the Korea Ministry of Land, Infrastructure and Transportation(MOLIT) as Innovative Talent Education Program for Smart City; and in part by the National Research Foundation of Korea (NRF) grant funded by the Korean Government (MSIT) under Grant 2022R1A2C1092737.

ABSTRACT This paper proposes a two-dimensional resource allocation technique for vehicle-to-infrastructure (V2I) communications. Vehicular communications requires high data rates, low latency, and reliability simultaneously. The 3rd generation partnership project (3GPP) included various numerologies to support this, leading to diversification of transmit time interval (TTI). It enables the two-dimensional resource allocation that considers time and frequency simultaneously, which has yet to be studied much. To tackle this issue, we propose a reinforcement learning approach to solve the two-dimensional resource allocation problem for V2I communications. A reinforcement learning agent in a base station allocates a quality of service (QoS) guaranteed two-dimensional resource block to each vehicle to maximize the sum of achievable data quantity (ADQ). It exploits received power information and a resource occupancy status as input. It outputs vehicles' allocation information that consists of a time-frequency position, bandwidth, and TTI, which is a solution to the two-dimensional resource allocation. The simulation results show that the proposed method outperforms the fixed allocation method. Because of the ability to pursue ADQ maximization and QoS guarantee, the proposed method performs better than an optimization-based benchmark method if each vehicle has a QoS constraint. Also, we can see that the resource the agent selects according to the QoS constraint varies and maximizes the ADQ.

INDEX TERMS Deep reinforcement learning, V2X communications, quality of service, resource allocation.

I. INTRODUCTION

With the advent of complex applications that combine high data rates, low latency, or high reliability, discussions on next-generation communication networks have been actively conducted to support them. The international telecommunication union (ITU) radiocommunication sector has defined three service types to meet the requirements of new applications: enhanced mobile broadband (eMBB) for applications requiring high data rates, massive machine-type communications (mMTC) for applications requiring high-density networks, and ultra-reliable low-latency communications (URLLC) for latency-sensitive applications. Vehicle-to-everything (V2X) communications is a highly complex application requiring all three service types. It consists of several vehicle-related communications, such as vehicle-to-infrastructure (V2I) and vehicle-to-vehicle (V2V) communications. Vehicles utilize V2V communications for direct information exchange between themselves and V2I communications to convey information to the infrastructure such as base stations or roadside units (RSUs) and vice versa [1], [2], [3], [4].

Early vehicular communications focused on collision avoidance to reduce car crashes. They need delay-sensitive

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

communications, such as V2V communications, to recognize the location of other vehicles as quickly as possible [5]. Dedicated short-range communications (DSRC) technology, a carrier sensing multiple access-based technique, was developed to support early V2V communications. Each DSRC-equipped vehicle broadcasts its driving information to let the others decide on potential risks of collisions. Other research for V2V communications also has been studied. In [6], V2V multi-hop modeling to increase the efficiency of information dissemination has been proposed. In [7], media access control using high frequency and periodicity of safety message broadcast has been proposed. In [8], the authors collected and analyzed actual DSRC data to model V2V communications in an urban environment and presented a reliable beaconing method. In [9], the authors have proposed novel in-vehicle network mobility models considering car-following and avoidance behavior. The models show a performance enhancement of the V2V communications in 5G NR networks. In [10], the authors have proposed a V2X resource allocation scheme using machine learning. They presented a joint power, spectrum, and local computing ratio allocation problem with partial CSI and offered a solution that minimizes V2I link delay and satisfies the V2V reliability constraint.

However, V2V communications has limitations in satisfying complex functions for recent vehicles. For autonomous driving, vehicles need to share data generated by their various sensors, so some methods supporting a high data rate and reliable information exchange between vehicles are required. Also, applications such as infotainment require a super high data rate and relatively small delay because the network needs to support video, augmented reality (AR), or virtual reality (VR) data transmission wirelessly for the vehicles. As such, vehicular communications must support short delay time, high data rate, and high reliability. It is challenging to meet these demands with V2V communications. Therefore, the vehicles need some aid from the high speed communication entities, such as base stations or RSUs.

Meanwhile, these entities need to be more flexible to support vehicular communications. The 3rd generation partnership project (3GPP) defined new numerology of orthogonal frequency division multiplexing (OFDM) to support new services [11]. Each numerology has different subcarrier spacing (SCS). Enabling various options of SCS stimulates the diversification of transmit time interval (TTI) options, resulting in flexible numerology allocation according to communications service type. For example, the network can allocate numerology with short TTI to reduce latency for a URLLC service or with small SCS for applications that need stability, such as mMTC. Also, various TTI options diversify resource block size and resource allocation flexibility, enabling two-dimensional resource allocation with resources that have different numerologies.

However, it must overcome the inter-numerology interference (INI) problem caused by out-of-band emission (OOBE) because resources with different SCS are not orthogonal. In [12], the authors have proposed an OFDM-based physical layer with scalable numerologies. They mentioned filters are mandatory to control it. In [13], the authors conducted a field test to compare the INI reduction performance of cyclically prefixed OFDM (CP-OFDM), windowing OFDM (w-OFDM), and filtered OFDM (f-OFDM). In [14], the authors analyzed the INI between users with different numerology and the performance of guard subcarriers to reduce INI. They also compared the INI suppression performance of CP-OFDM and w-OFDM. In [15], the authors analyzed users' performance with different numerology and sampling rates. The papers above say that the guard subcarriers significantly reduce INI. If appropriate numbers of guard subcarriers are used, users with different numerology can be simultaneously serviced in one band or subband, leading to two-dimensional time-frequency resource allocation using multi-numerology.

Since most standards and communication methods exploit a fixed TTI, the two-dimensional resource allocation is relatively unique research content. There are two popular ways to allocate two-dimensional resources, one is heuristic approaches, and the other is optimization-based approaches. Optimization-based approaches mainly use metrics such as sum-capacity, latency, quality of service (QoS), reliability, energy efficiency, et cetera. In [16], the authors have proposed a heuristic solution to the time-frequency resource allocation problem to optimize allocation efficiency in situations with/without QoS constraints. In [17], a heuristic technique has been proposed to determine the size of resource blocks based on the allocation proposal and optimize the protocol efficiency and allocation efficiency. Also, a time-frequency resource allocation method considering multi-numerology has been actively studied these days. In [18], which is our benchmark method, the authors have proposed a method that satisfies the QoS of users with strict latency while maximizing the capacity of high throughput users when there are various service requests from users in a multi-numerology environment. They proved the problem is NP-hard and presented a sub-optimal, low-complexity solution based on linear programming relaxation and Lagrangian dual. However, it has an unbalanced resource problem for a slow fading channel and a case of all high throughput users. It also shows an inability to simultaneously pursue capacity maximization and QoS guarantee in the case of all QoS users. In [19], the authors have proposed a two-dimensional resource allocation that maximizes energy efficiency in services with heterogeneous latency requirements. The authors have improved energy efficiency by adding an algorithm that turns the power amplifier on and off in response to traffic changes.

Although there are such results on the two-dimensional resource allocation considering multi-numerology, there are still many problems related to QoS and complexity. Recently, reinforcement learning (RL) has been widely used to solve high-complexity and difficult-to-solve problems. RL is a method to determine the action that maximizes the return, which is a weighted sum of the rewards from the moment of the action selection until the end of the episode, and it can maximize not an instantaneous reward but a return that considers all of the rewards thanks to this attribute. The RL agent keeps updating a table or function approximator of the expected return, which is the value function, during the training. The action selected by the trained agent maximizes the return on average, and the designer can decide how many time steps the agent considers by controlling the constant of the weighted average. In short, the RL agent chooses the action that produces the most significant profit for the agent on average, which motivates the use of the RL in this research.

These attributes make RL perform well for complex problems [20], [21]. In [22], the authors have proposed solutions for power allocation and spectrum sharing problems in unicast and broadcast V2V networks using RL. In particular, the authors improved performance by including latency as well as the sum-rate of V2I and V2V links in reward. In [23], energy-efficient user scheduling and resource allocation techniques with a hybrid energy source have been proposed. The authors analyzed the performance of RL techniques and energy efficiency with and without a sustainable energy source. In [24], the authors have proposed a numerology selection and spectrum allocation policy to maximize the aggregated capacity of mobile virtual network operators (MVNO). The authors presented the performance according to the number of sub-bands and MVNOs and compared the performance of the deep RL (DRL)-based method with the optimal algorithm. In [25], the authors have proposed an RL-based Max-Min sum Rate enhancement method. They present an optimization problem that outputs an optimal three-dimensional position of an unmanned aerial unit (UAV) and power allocation. Because of the expensive computational cost, the authors exploited Q-learning with an exploitation strategy. The method showed a better performance than the conventional waterfilling scheme. In [26], the authors have proposed a decentralized resource allocation using multiagent double deep Q learning. Their method aims to maximize the V2I sum-capacity while offer reliable communication for V2V links. In [27], the authors have proposed a DRL-based resource allocation scheme that outputs each vehicle's resource blocks and transmit power. The DRL agent maximizes the sum-capacity while reducing the power consumption of the V2V links. In [28], the authors designed a novel fog computing framework for a smart healthcare system. They presented a structure for data exchange for the healthcare system. After that, they exploited Q-learning for the diagnosis of the patients. In [29], the authors presented DeepMist architecture, which is a low latency energy efficient deep learning-based mist computing structure for managing healthcare big data. They incorperated reinforcement learning for prediction of the heart disease.

To sum up, the two-dimensional resource allocation still has problems like the relatively inefficient performance of the heuristic approaches and the computational invalidity of optimization-based approaches, and RL is a perfect match to tackle that. Therefore, we propose an efficient RL-based two-dimensional resource allocation method in V2I communications. Our main contributions are as follows:

- We propose the achievable data quantity (ADQ) as a metric for solving two-dimensional resource allocation with multiple TTIs and bandwidth (BW). The capacity is a suitable metric for resource allocation considering a single TTI. However, in the case of having multiple TTIs, we need a metric to reflect the impact of diversified TTIs. Therefore, we propose the ADQ as a metric for the efficient resource allocation method in case of having multiple numbers of BW and TTI.
- Thanks to the RL method that maximizes the expected return, RL performs better than the other optimization-based scheme, motivating us to use it for the research. We propose a DRL formulation to solve the problem of two-dimensional resource allocation with multiple TTIs. The agent exploits the received signal power of each vehicle and resource occupation state as its state and outputs an action with a time-frequency position, bandwidth, and TTI. We design a reward that reflects ADQ, QoS constraint, and overlapping. With the trained agent, the network produces the maximum sum ADQ, and every vehicle can use non-overlapping resources that guarantee a QoS constraint.
- We simulated the proposed method to evaluate the method's performance. We also simulated the optimization-based benchmark method for comparing the performance. When there is a QoS constraint, the results show that the agent allocates more resources to vehicles with poor link quality than no QoS constraint environment, which says the proposed method changes its policy according to the QoS. Also, the proposed method can maximize a sum ADQ and guarantee a QoS constraint of each vehicle simultaneously, which the benchmark does not have.

The rest of the paper is organized as follows. In Section II, we introduce the system model, problem formulation. In Section III, the RL formulation to solve the proposed problem is described. In Section IV, we show the simulation results and draw conclusions in Section V.

II. PRELIMINARIES

A. SYSTEM MODEL

The network has a macro base station (MBS) and K vehicles. The set of vehicles is $\mathcal{K}$. An RL agent in the MBS allocates a resource block to each vehicle based on its received power and the occupation state of the resources. The agent's goal is to maximize the sum of ADQ with/without QoS constraint. Fig. 1 describes the DRL-based method and the environment that includes it.

The network considers two-dimensional resource allocation. It has a resource space, as shown in Fig. 2. The time duration of the resource space is T, and the frequency range is B. The numerology specifies the TTI of the resource block.

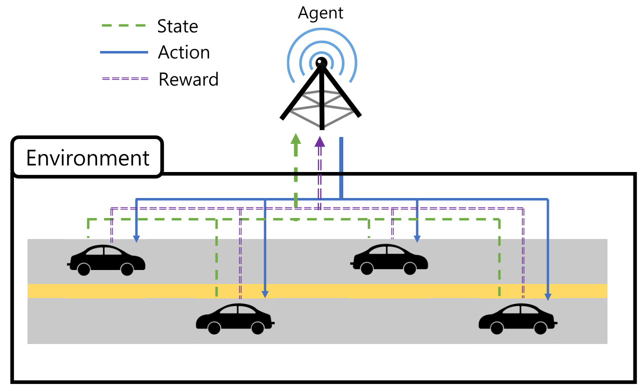

FIGURE 1. The DRL-based method and the environment.

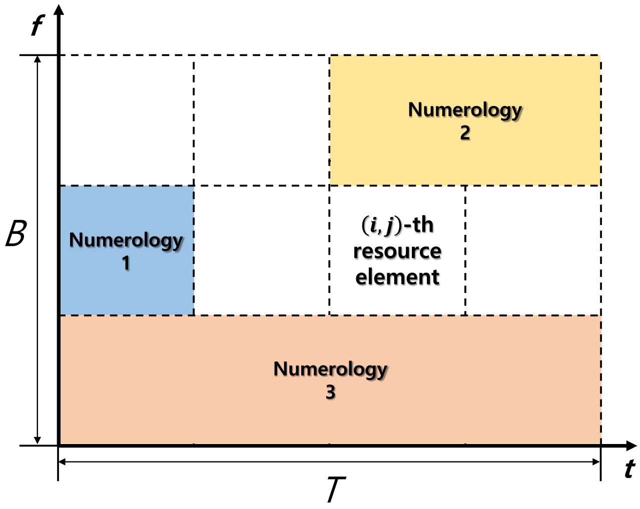

FIGURE 2. Resource block of the system.

When the numerology of vehicle $k \in \mathcal{K}$ is $\mu_k \in \mathcal{M} = \{0, 1, \ldots, \mu^{max}\}$, the TTI of the vehicle k is $T_k = 2^{-\mu_k} T^{max}$, where T^{max} is the TTI for $\mu = 0$. The bandwidth of the vehicle k is $B_k \in \mathcal{B} = \{B^1, B^2, \ldots, B^b\}$.

We partition the resource space into resource elements. The time duration of it is T^{min}, and BW is B^{min}. T^{min} is the TTI for μ^{max}, and B^{min} is the smallest BW. We name the resource element i-th in time and j-th in frequency as (i, j)-th element. We denote the TTI and BW for vehicle k divided by T^{min} and B^{min} as T_k^e and B_k^e, respectively. T_k^e and B_k^e mean that how many resource elements is needed to express T_k^e and B_k^e, respectively. The MBS allocates a resource block with horizontally or vertically consecutive resource elements to each vehicle in the resource space.

The interference from other vehicles to vehicle k is

$$I_k = \frac{1}{T_k} \sum_{k' \neq k} \sum_{i,j} \rho_{k,ij} \rho_{k',ij} \frac{P_{k'} \tilde{h}_{k'}}{B_{k'}} B^{min} T^{min}, \tag{1}$$

where $P_{k'}$ is the transmit power of vehicle k', $\tilde{h}_{k'}$ is channel power gain of the interference from vehicle k', $\rho_{k,ij} \in \{0, 1\}$ indicates whether vehicle k occupies the (i, j)-th element. Note that we calculate the interference energy in each resource element, then sum it all up and divide it by T_k to get the interference power. The signal-to-interference-plus-noise ratio (SINR) of vehicle k is

$$\gamma_k = \frac{P_k h_k}{N_0 B_k + I_k}, \tag{2}$$

where h_k is the power gain of the channel, and N_0 is noise spectral density. The ADQ of vehicle k is

$$D_k = B_k T_k \log_2(1 + \gamma_k). \tag{3}$$

Note that we use D_k instead of capacity, which is an appropriate metric if the T_k of each user is the same as in the existing resource allocation. However, D_k is greatly affected by diversified T_k, which cannot be reflected if we use the capacity instead of D_k.

B. PROBLEM FORMULATION

The optimization task is to allocate resource blocks with consecutive resource elements to vehicles to maximize the sum of D_k while guaranteeing a QoS constraint for each vehicle. The optimization problem is

$$\begin{aligned}
&\max_{(\rho_{k,ij})} \sum_{k \in \mathcal{K}} D_k \\
&\text{s.t.} \sum_{i=i_s}^{i_s+T_k^e-1} \rho_{k,ij} = T_k^e,\ j \in \{j_{k,s}, j_{k,s+1}, \ldots, j_{k,f}\}, \quad \forall k \in \mathcal{K} \\
&\quad\ \sum_{j=j_s}^{j_s+B_k^e-1} \rho_{k,ij} = B_k^e,\ i \in \{i_{k,s}, i_{k,s+1}, \ldots, i_{k,f}\}, \quad \forall k \in \mathcal{K} \\
&\quad\ D_k > D_{min}, \quad \forall k \in \mathcal{K},
\end{aligned} \tag{4}$$

where $i_{k,s}$ and $i_{k,f}$ are the starting and finishing position of the allocated resource block for vehicle k in time, respectively, $j_{k,s}$ and $j_{k,f}$ are the starting and finishing position of the allocated resource block for vehicle k in frequency, respectively, and D_{min} is a QoS constraint. The first two constraints mean resource blocks allocated to each vehicle are continuous, and the last one is about the QoS guarantee.

C. REINFORCEMENT LEARNING

RL is a method of learning a policy, which is the selection probabilities of actions, by interacting with the environment [30]. The agent chooses an action in a state and receives a reward from the environment for taking action in the state. It is crucial to appropriately design the state, action, and reward to control the agent properly. To understand why, we need to know about the Markov decision process (MDP).

MDP provides a mathematical foundation for RL. It models a decision-making process with probability and Markov property. The next state is determined stochastically depending on the present state and action. The agent receives a reward, which is an evaluation of the action. The return is a discounted sum of the rewards.

$$\begin{aligned} G_t &= \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \\ &= R_t + \gamma G_{t+1}, \end{aligned} \tag{5}$$

where $0 \leq \gamma \leq 1$. We use the expected return, which is the value function, as a metric to maximize,

$$q_\pi(\boldsymbol{s}, \boldsymbol{a}) = \mathbb{E}_\pi \left[G_t \middle| S_t = \boldsymbol{s}, A_t = \boldsymbol{a} \right], \tag{6}$$

where π is a policy, $\boldsymbol{s} \in \mathcal{S}$ is a state from the state space, $\boldsymbol{a}$ is an action from the action space, S_t is a state sampled at time step t, and A_t is an action sampled at time step t. As we can see from the equation (6), $q_\pi(\boldsymbol{s}, \boldsymbol{a})$ can be different as the policy π changes, and the purpose of the agent is to learn a policy that maximizes the $q_\pi(\boldsymbol{s}, \boldsymbol{a})$,

$$\pi(\boldsymbol{s}) = \arg\max_{\boldsymbol{a}} q_\pi(\boldsymbol{s}, \boldsymbol{a}). \tag{7}$$

There are many algorithms to solve (6) and (7), but we have to know the exact state transition probability and reward distribution, which is too hard to know in many cases [31]. RL provides a breakthrough to get (6) without additional information. The RL agent chooses an action in a state and receives a reward for taking action. Then, it is possible to sample (6) and update the agent's policy using the rewards. The detailed techniques of RL can be distinguished by how to update (6), whether to use stochastic policy, and whether to use the function approximation using various methods such as deep learning (DL) or not, et cetera.

The tabular RL method, which updates a table for value function every time the agent visits a state-action pair, has a dimensionality problem. A larger number of states and actions makes the agent explore more because there is no way to fully update the value function without visiting all state-action pairs. This causes a severe problem of long training and test time.

The function approximation method is a way to solve this problem [30]. This learning-based method expresses the value function with features and weights. Several methods exist, such as linear approximation and tile coding, but deep RL (DRL) has been the most popular recently. The DRL method exploits the DL model to estimate the value function. The agent has a DL model trained with a loss function based on a difference between G_t and $\hat{q}(s, \boldsymbol{a}, \boldsymbol{w})$, which is an approximated value function by the DL model. With powerful performance and many kinds of research proving DRL's efficiency, DRL makes RL much more popular than ever.

III. PROPOSED METHOD

This section introduces a two-dimensional resource allocation method for V2I communications using DRL, which we propose. We describe the state, action, and reward for DRL to solve the problem. Finally, the training and testing processes are described.

A. STATE

An RL agent gets information about its state from the environment. We assume that states are a deterministic function of observations. Designing states properly is essential in RL because of its influence on learning time and performance. We define the state for this problem with four elements, the power of the received signal from the vehicle k, that of $k+1$, the occupation state of the resource space, and the terminal state indicator.

We use the power of the received signal from the vehicle k, $P_k^r = P_k h_k$, as the first state element when allocating the resource block of vehicle k. P_k^r is an essential part of the (2), significantly affecting the ADQ. It helps the agent to guess how much ADQ the link between the MBS and the vehicle would produce. It also allows the agent to allocate the appropriate resource block size depending on the QoS constraint. We note that we take log base ten to P_k^r because of its indistinguishably small value, so the first state element is $s_1 = \log_{10} P_k^r$.

The second state element is the received power of vehicle $k+1$, P_{k+1}^r, which will be allocated right after the vehicle k. Every time the agent allocates a resource block to each vehicle, the agent needs to know the received power of other vehicles to guess how large P_k^r is among the vehicles. So we choose to exploit P_{k+1}^r as the second state element to give the agent contextual information about the received power of other vehicles, which helps to decide the resource block size of the vehicle k. Note that the agent cannot distinguish each vehicle if we use all the received power from the vehicles as the second state element. Doing it makes a similar input state every time the agent gives a resource to each vehicle, which is meaningless. We also note that the second state element is 0 if there is no more vehicle to allocate after vehicle k, and we take log base ten if there are vehicles to allocate. The second state element is

$$s_2 = \begin{cases} 0 & \text{if the state is terminal} \\ \log_{10} P_{k+1}^r & \text{otherwise.} \end{cases} \tag{8}$$

The third state element is the occupation state of the resource space. The agent needs this to avoid allocating overlapped resources to each vehicle, which leads to SINR degradation. The $\rho_{ij} \in [0, 1, \ldots, K]$ denotes how many vehicles occupy the (i, j)-th element during the allocation process. The third state element is

$$\boldsymbol{s_3} = [\rho_{00}, \rho_{01}, \ldots, \rho_{0q}, \rho_{10}, \rho_{11}, \ldots, \rho_{pq}], \tag{9}$$

where $p = T/T^{min}$ and $q = B/B^{min}$ are the last time and frequency index of the resource space expressed as a resource element, respectively.

The fourth state element is an indicator,

$$I_t = \begin{cases} 1 & \text{if the state is terminal} \\ 0 & \text{otherwise,} \end{cases} \tag{10}$$

that lets the agent know whether it is a terminal state or not. We need this because there is no particular expression of terminal state like the stage clear in game playing problem in the simulator. Overall, we use four state elements for state vector for vehicle k as

$$\boldsymbol{s}^k = [s_1, s_2, \boldsymbol{s_3}, I_t]. \tag{11}$$

B. ACTION

It is vital to design suitable action to solve the problem and adequately control the agent. Since we are working on the two-dimensional resource allocation problem, setting the time-frequency position, TTI, and BW of the resource block is the appropriate action for the agent. Therefore, the action of the vehicle k is

$$\boldsymbol{a}^k = [f_k, t_k, B_k, T_k], \tag{12}$$

where f_k and t_k are the position of the resource element in the resource space. The first two elements determine the starting point of the resource block, and the last two elements determine its size.

C. REWARD

The reward is the evaluation of actions chosen by the agent. The agent in a state has the value function and updates it with rewards, then utilizes it to decide what action to take if it faces the same state again. It is vital to design rewards properly to reinforce the agent as we want. We propose one positive and two negative reward elements for controlling the agent. Using the combination of the three reward elements, we propose the reward for training agents with/without the QoS constraint.

The first reward element is the ADQ at the moment of the allocation. The agent trained with this reward element tries to give vehicles with high channel quality a large resource block and with low channel quality a small resource block. The first reward element is

$$\begin{aligned} r &= c_1 D_k \\ &= c_1 B_k T_k \log_2(1+\gamma_k), \end{aligned} \tag{13}$$

where c_1 is a constant. We can guess two consequences with r. The first is SINR improvement. If multiple vehicles use the overlapped resource, the log part of D_k will be significantly reduced, leading to a decrease in r. Therefore, the agent would allocate each vehicle's resource block in the empty part of the resource space so that interference does not occur. The second is fully exploited resource space. The agent will try to fill the resource space as much as possible because D_k increases with $B_k T_k$. In other words, D_k prevents the resource space from being empty. However, if the received power of some vehicles is much larger than the other vehicles, (13) is large by itself, so the agent learns to allocate most of the resources to the vehicles. In this case, certain vehicles occupy most of the resources, hindering the resource allocation of others. For this reason, other rewards should be exploited to prevent this situation.

Note that r is derived immediately after the agent executes an action, regardless of what resource block the agent allocates to other vehicles. Once r is determined, it is not changed even if overlaps happen after the allocation processes of other vehicles, leading to a decrease in the value. If we calculate r after finishing resource allocation for all vehicles and use it as a reward element, the MDP is broken because the agent's subsequent actions affect the reward.

The second reward element is a negative reward for allocating overlapped resources. The agent trained with this has some ability to allocate a non-overlapping resource to each vehicle. It can also control allocating a manageable size of each resource because too large resources make overlaps because of the little resource space to allocate.

The third reward element is a negative reward for the QoS violation. The agent receives it if the ADQ of the vehicle is smaller than the pre-defined threshold, QoS constraint. The agent needs the capability to infer that the non-allocated part of the resource space is enough to guarantee QoS for others, and this reward element helps it. If we only use the reward elements mentioned earlier, the agent leaves a small part of resource space insufficient to satisfy the QoS constraint of each vehicle with bad channel quality. That could cause a severe problem if a vehicle with low channel quality needs to communicate in an emergency.

We use three reward element mentioned earlier for the reward. The reward, which maximizes the sum ADQ of all vehicles while guarantees the QoS, is

$$R_k^{cap} = \begin{cases} -c_2 & O_k \geq 1 \\ -c_3 & O_k < 1 \text{ and } D_k \leq D_{min} \\ r & O_k < 1 \text{ and } D_k > D_{min}, \end{cases} \tag{14}$$

where O_k means how many resource elements of vehicle k are overlapped. Note that, as can be seen from (14), we give the second reward a higher priority than the third reward because it is difficult to match the QoS constraint when an overlap occurs.

D. RL METHODS

RL varies greatly depending on the methods. They produce differences in the convergence of the value function estimate or in its representation. We mention three representative categories. The first is how to express the sample of the value function, the second is ϵ-greedy policy for exploration, and the third is the value function expression.

The Monte Carlo method is a way to sample the value function. A sample of the value function when using the Monte Carlo method is (5), and this is an unbiased estimator of (6). If we have enough samples, the value function estimate converges to the true value function. On the other hand, the learning process with Monte Carlo method is slow because the agent has to wait until the end of an episode, and it cannot be used in continuing tasks where the episode continues. However, the two problems of the Monte Carlo method do not make an impact in our case because we use relatively short episodes, which is not continuing. So, in our case, using the Monte Carlo method allows us to have an estimate with no bias and is not slow.

Algorithm 1 ϵ-Greedy Policy

```
Input: Value function $q_\pi$, state $s^k$, $\epsilon$.
Execution:
if $n_u \le \epsilon$
Generate uniform random variable $n_u \sim \mathcal{U}[0, 1)$.
if $n_u \le \epsilon$
    $p(a^k|s^k) = 1/|\mathcal{A}|$.
    Sample $a^k$ from $p(a^k|s^k)$.
else
    $a^k = \arg\max_{a} q_\pi(s^k, a)$
Return: $a^k$
```

TABLE 1. Simulation parameters.

Parameters	Value
MBS antenna height	24 m
Vehicle antenna height	5 m
Number of antennas of MBS and vehicle	1
Antenna gain of MBS and vehicle	0 dBi
Carrier frequency	2 GHz
Vehicle transmit power	23 dBm
MBS noise figure	5 dB
Noise spectral density	-173.93 dBm/Hz
Pathloss Model	TR 38.901 RMA LOS [32]
BW candidate (MHz)	[5, 10, 15, 20]
TTI candidate (ms)	[0.25, 0.5, 1]
μ_k	[0, 1, 2]
Number of episode	500,000
Greedy period T_{greedy}	400,000

TABLE 2. Guard band.

SCS	5 MHz	10 MHz	15 MHz	20 MHz
15 kHz	242.5 kHz	312.5 kHz	382.5 kHz	452.5 kHz
30 kHz	505 kHz	665 kHz	645 kHz	805 kHz
60 kHz	N/A	1010 kHz	990 kHz	1330 kHz

It is essential to balance exploration and exploitation in RL. Too much exploitation makes the agent greedy and prevents it from visiting all state-action pairs. Conversely, excessive exploration causes the agent not to be greedy, which prevents it from having a good policy. So we use ϵ-greedy policy to solve the problem. The policy is shown in **Algorithm** 1 [30].

In order to decide whether to express the value function with the tabular method or the function approximation, it is important to carefully consider the size of the state space and action space. When the sizes of the spaces are small, it is not easy to use the function approximation because of performance degradation by over-generalization, which makes states non-distinguishable. On the other hand, if the tabular method is used when the size of the spaces is large, the agent cannot complete the exploration itself, so learning is impossible. In our problem, the state space is relatively small compared to problems in other fields which use RL. However, in the case of the action space, if the size of the resource elements is 0.25 ms and 5 MHz in our simulation and that of the resource space is 2 ms and 20 MHz, the number of resource elements is 32. Considering TTI, BW, and resource location, the number of possible actions is more than 150. That is a lot considering the number of actions in the existing problem in other fields [33]. Therefore, we use function approximation, which is trained using the Huber loss function for vehicle k, which is

$$L_k = \begin{cases} 0.5(\hat{q}(s, a, w) - G_t^k)^2 & \text{if } |\hat{q}(s, a, w) - G_t^k| \le 1 \\ |\hat{q}(s, a, w) - G_t^k| - 0.5 & \textit{otherwise}. \end{cases} \tag{15}$$

E. TRAINING AND TEST

The proposed method has training and testing algorithms like other RL algorithms. At the start of the training algorithm, as shown in **Algorithm** 2, we initialize the neural network and the episode buffer. After that, we repeat the following process for each episode. First, we generate vehicles, MBS, and links between them. Second, the agent samples a vehicle's state, chooses a action according to the policy, and receives a reward. It is stored in the episode buffer. After completing the episode, the neural network is updated based on the return calculated from the rewards in the buffer. We decay ϵ at the end of every episode for the ϵ-greedy policy. When ϵ_{max} is 1, ϵ_{min} is 0.1, and the greedy interval is T_{greedy}, the ϵ is

$$\epsilon_t = \min(\epsilon_{t-1} - \frac{\epsilon_{max} - \epsilon_{min}}{T_{greedy}}, \epsilon_{min}). \tag{16}$$

Note that we need to wait until the episode is over since we use the Monte Carlo technique, so we use a buffer to store each data. The test algorithm shown in **Algorithm** 3 is the same as the training algorithm, except that there is no training of the neural network and no buffer.

IV. SIMULATION

In this section, we describe the simulation environment discuss the simulation results.

A. ENVIRONMENT AND NEURAL NETWORK ARCHITECTURE

It is crucial to make a concrete simulation environment that models a realistic environment as possible. Fortunately, some standards offer information about the communication channel and how to deploy the entities, such as an MBS and vehicles. We refer to the freeway case of 3GPP TS 36.885 for this purpose [34]. We assume a slow fading and frequency flat channel. The road configuration is 2 km long and 4 m wide in one lane, and six such lanes exist. When the horizontal position is x, the vertical position is y, and the antenna position is z, the position of each object on the road is (x, y, z). The MBS is at (999.5m, 0m, 35m), and each vehicle is located at (u m, v m, 1.6m), where $u \sim U(0, 2000)$ and v is the discrete uniform distribution that outputs a value from a set $\mathcal{V} = \{2, 6, 10, 14, 18, 22\}$. The size of the resource space is 2 ms-20 MHz. Please refer to Table 1 for the details about the simulation environment.

Table 2 shows the guard band size [11]. We assume there is no INI since the guard band is large enough [13]. Assuming

Algorithm 2 Training

```
Input: The parameters of the network.
Output: The weights of the trained network $\theta^{\mu}$.
Training:
Initialize the neural network.
Initialize the episode buffer $\mathcal{D}$.
for each episode $t = 0, 1, \ldots, T$ do
    Place vehicles and an MBS on the simulation plane.
    Initialize links between the MBS and vehicles.
    for each vehicle in the episode $k = 0, 1, \ldots, K$ do
        The agent samples $s^k$, the state of vehicle $k$.
        The agent selects an action based on the $\epsilon$-greedy policy, $a^k$.
        Execute action, receive reward, $R$.
        Save the tuples $(s^k, a^k, R)$ in $\mathcal{D}$.
    end for
    Calculate all of the return of vehicles by using the data saved in $\mathcal{D}$.
    Update the neural network by minimizing the loss in (15).
    Clear $\mathcal{D}$.
    $\epsilon \leftarrow \min(\epsilon - (\epsilon_{max} - \epsilon_{min})/T_{greedy}, \epsilon_{min})$.
end for
Return: $w$
```

Algorithm 3 Test

```
Input: The trained neural network $w$.
Test:
Load $w$.
for each episode $t = 0, 1, \ldots, T$ do
    Place the vehicles and the MBS on the simulation plane.
    Initialize links between MBS and vehicles.
    for each vehicle in the episode $k = 0, 1, \ldots, K$ do
        The agent samples $s^k$, the state of vehicle $k$.
        The agent selects action, $a^k$.
    end for
    Evaluate the actions.
end for
```

that the guardband is divided into equal sizes and located at both ends, each contains eight or more subcarriers, which is sufficient. Note that we do not use the 5 MHz BW for $\mu = 2$, as in 3GPP TS 38.101-1 and the Table 2 [11]. We consider the cyclic prefix length as defined in [35].

The neural network of this research consists of 2 layers, the first layer has 1024 nodes, and the second layer has 168, equal to the number of actions. The activation function of the first layer is the Rectified Linear Unit (ReLU). The learning rate is 2×10^{-4}. We use L1 and L2 regularizer as kernel regularizers, and their coefficients are 10^{-5} and 10^{-4}, respectively.

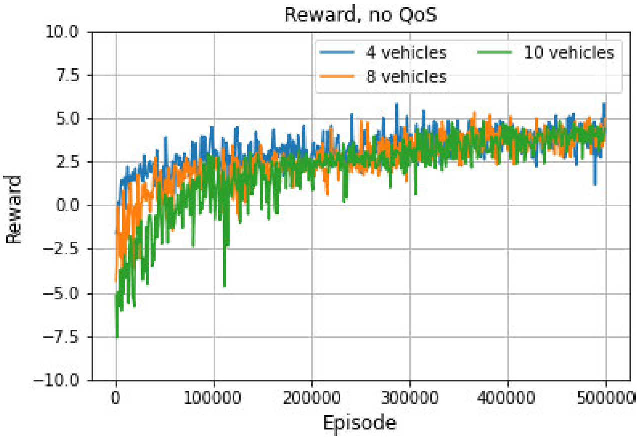

FIGURE 3. Training reward progress (D_{min} = 0).

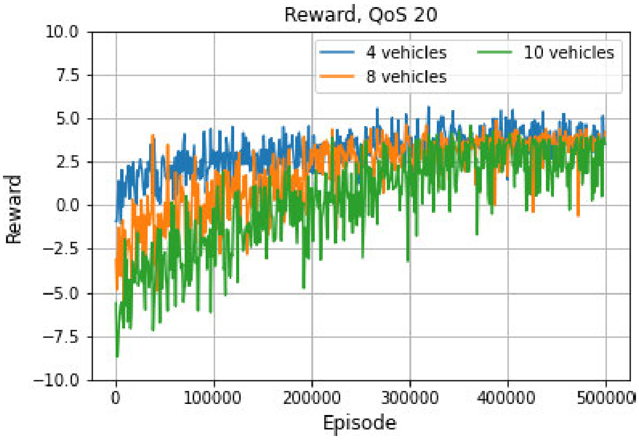

FIGURE 4. Training reward progress (D_{min} = 20 kbit).

B. SIMULATION RESULTS

In this subsection, we present the results of the simulation. Fig. 3 and Fig. 4 show the reward according to the training progress with/without a QoS constraint. D_{min} in the Fig. 3 is 0 and in the Fig. 4 is 20 kbit. In all cases, the agents for each constraint find an excellent policy since the reward increases as the training progresses. At the beginning of the training, the rewards decrease as the number of vehicles increases because there is a high probability of resource overlaps when using a poorly trained policy. The rewards at the end of the training show us the policy is converged well.

Fig. 5 shows the performance of the agents trained with different rewards. We tested this 100,000 times. In the figure, NoQoS is when there is no QoS constraint, QoS20 is the case where there is a reward element for overlap and a QoS constraint of 20 kbit. NoOL20 is the case with the QoS constraint and no negative reward element. NoNeg is the case where there are no negative reward elements. Detailed constants are in the Table 3. Fig. 5 verifies our reward design. The NoNeg case shows the worst performance among the agents, and the NoOL20 case is better than that, but there is still much room for performance improvement. The NoQoS case

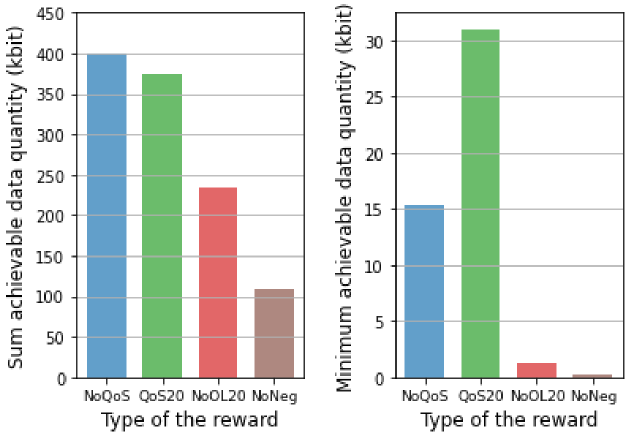

FIGURE 5. Comparison of the proposed agents with the different reward (8 vehicles).

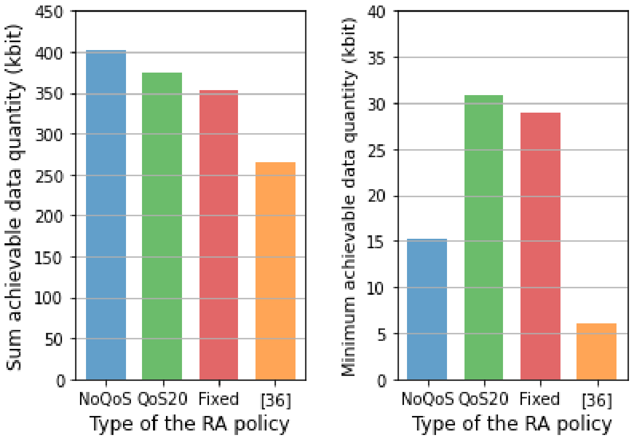

FIGURE 6. Comparison of the proposed and fixed RA methods (8 vehicles).

TABLE 3. Constants of the reward for each case.

Cases	c_1	c_2	c_3
NoQoS	0.01	1	0
QoS20	0.01	1	1
NoOL20	0.01	0	1
NoNeg	0.01	0	0

shows the best sum ADQ performance among the agents but relatively bad minimum ADQ performance. The QoS20 case shows relatively high sum ADQ performance and the best minimum ADQ performance. This means that the negative reward element for violating a QoS constraint reinforces the agent to make actions to reserve resources for the vehicles with bad channel quality.

Fig. 6 shows the performance of the RL-based methods, the resource allocation with a fixed size policy, which is a cellular's policy, in the environment with eight vehicles, and the method in [36]. The resource allocation with fixed size policy allocates resources of the same size to all vehicles without overlap. The bandwidth and TTI of the resource

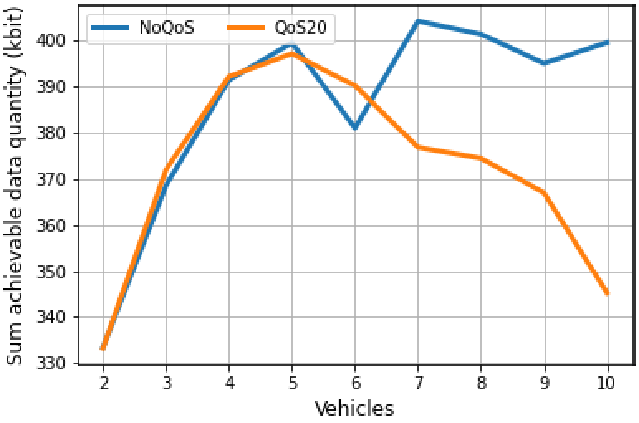

FIGURE 7. Sum ADQ of the proposed agents as the number of vehicles increases.

allocated using this method are 10 MHz and 0.5 ms, respectively. The method in [36] is a resource allocation strategy to service eMBB and URLLC users at the uplink simultaneously. In this method, the basestation allocates fixed resources to eMBB users and forces the eMBB users to share their resources with URLLC users. For comparison, each eMBB user has 10 MHz-1 ms resources and shares their resources by puncturing or superposition with a URLLC user, and they should decide how to share the resources. Each URLLC user gets 10 MHz-0.25 ms resources from the eMBB user. One of the proposed agents, one without QoS constraint, shows higher sum ADQ performance compared to the case of using fixed policy but lower minimum ADQ performance. However, the agent with 20 kbit QoS constraint shows good performance in all cases compared to the fixed policy. This means that the proposed agents are superior to the fixed policy. Moreover, the proposed agents show superior performance than the method in [36], and this is because of the number of users and the flexibility of the resource allocation method. The superposition method in [36] exploits non-orthogonal multiple access, which performs great when there is a large difference in signal power. So, one user of the eMBB-URLLC pair must be nearby the basestation, and the other must be at the edge of the cell. However, in the target environment, a highway vehicular communication environment, there are a few users, which means there is no guarantee of a significant distance between each user. Also, the proposed methods have flexibility in deciding the resource size of each user, which has the advantage of maximizing ADQ and guaranteeing QoS.

Figs. 7 and 8 show the sum ADQ and the minimum ADQ of the agents trained with the proposed method among the vehicles, respectively. We ran 100,000 simulations for each vehicle case. In the case of the agents with a QoS constraint in Fig. 7, the sum ADQ starts to decrease when there are more than 5 vehicles because the agent allocates many resources to vehicles with bad channel quality to guarantee QoS. This also can be seen in Fig. 8. For agents with a QoS constraint,

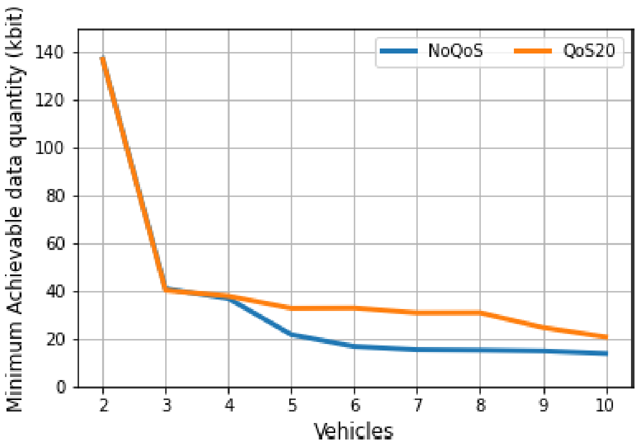

FIGURE 8. Minimum ADQ of the proposed agents as the number of vehicles increases.

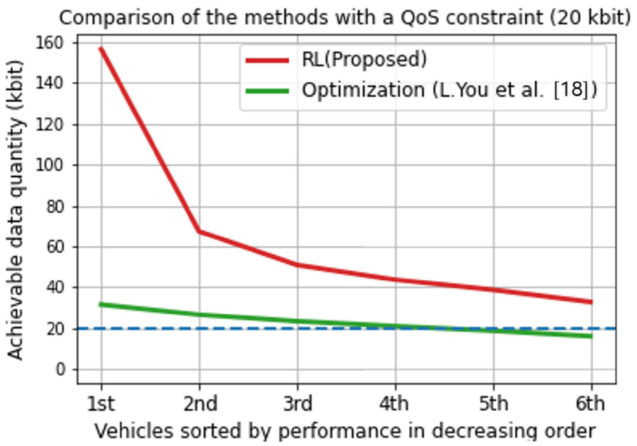

FIGURE 10. The ADQ of users sorted in descending (6 vehicles, $D_{min} = 20$ kbit).

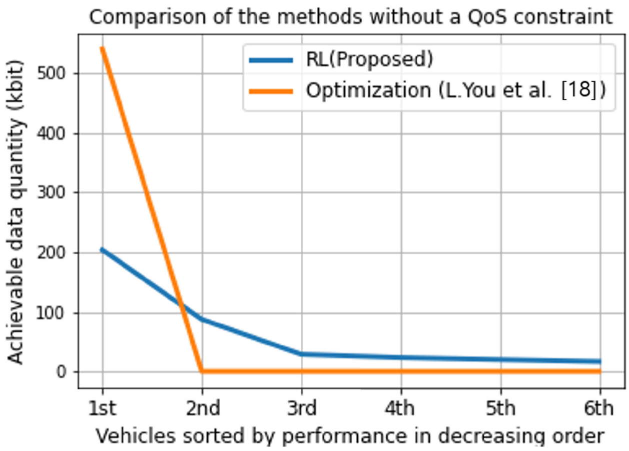

FIGURE 9. The ADQ of users sorted in descending (6 vehicles, $D_{min} = 0$).

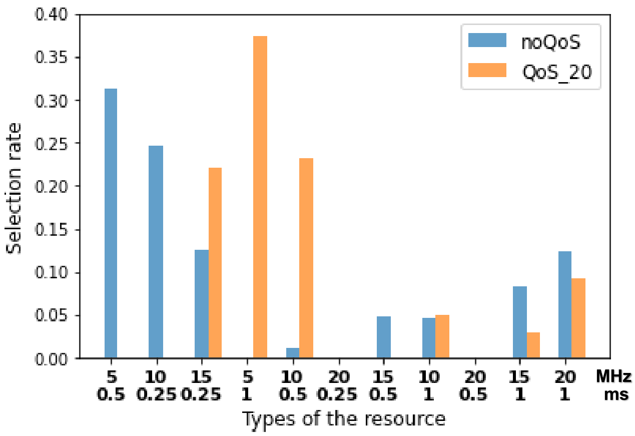

FIGURE 11. Selection rate of each resource.

the minimum ADQ among the vehicles is larger than the case without a QoS constraint when there are more than 5 vehicles. We conclude from these results that the reward element for a QoS violation makes the agent act differently and forces it to reserve resources for users with bad channel quality.

Figs. 9 and 10 show performance comparisons of the proposed method and the existing method [18] with and without a QoS constraint, respectively. The constants for Fig. 9 are same with the NoQoS case, and those for Fig. 10 are same with the QoS20 case. We simulated all the methods 10,000 times and averaged the results. Fig. 9 shows us that the best performance user in the benchmark method is better than the one in the proposed method, but the others in the benchmark have no resources. The benchmark method allows the MBS to allocate multiple resource blocks to each user. That means allocating all the resources to the user with the best channel quality is the most suitable way to get the highest ADQ. However, there is no resource to allocate to other users. Even if the purpose is to get the sum ADQ as high as possible, it is irrational for most of the users. Fig. 10 shows a comparison of the benchmark and the proposed method with a QoS constraint of 20 kbit, and we can see that the proposed one outperforms the benchmark method. The benchmark method allocates resources to QoS users as small as possible and the rest of the resources to users requiring a large volume of data. This causes a problem of the underutilized resource space if there are only QoS users. However, unlike the benchmark method, the proposed one pursues to maximize ADQ and satisfy the QoS constraint simultaneously and shows higher ADQ performance and no QoS violation.

Fig. 11 shows the probability that each resource block is selected by the agents with/without a QoS constraint. Cases with 2 to 10 vehicles were simulated 100,000 times each, and the number of times each resource block is selected is added up and then divided by the number of resource selections. We place the resources in the order from smallest to largest in the Fig. 11. Two things can be checked from this figure. The first is that if the number of resource elements is the same, the agent allocates a resource to each vehicle with the resource elements long arranged in time. This occurs because of the shape of the resource space. In our simulation environment, the number of resource elements in time is more than those in frequency. That means the agent trained by the proposed RL-based method has learned how to use resource space efficiently. This will have greater meaning in resource allocation methods with delay constraints or semi-persistent resources. The second is that the QoS constraint increases the size of the resource selected by the agent for a vehicle with bad channel quality. Resource blocks composed of two resource elements are the smallest resource. The vehicles with these resource blocks often have bad channel quality if there is no QoS constraint to maximize ADQ. However, if there is a QoS constraint, the agent must allocate a resource block of enough size to the vehicle to ensure QoS.

V. CONCLUSION

In this paper, we have proposed an RL-based two-dimensional resource allocation technique with/without QoS for V2I communications. We formulated a two-dimensional resource allocation problem that considers multiple TTIs and BWs, maximizes sum ADQ, and satisfies several constraints including QoS. We have proposed a DRL formulation to solve the problem of two-dimensional resource allocation. The state is the received power of vehicles and the occupancy of the resource space, and the action is the resource start time, start frequency, and size of the resource. In the reward design process, we found that capacity is not a function of the TTI and does not fit to train the RL agent for this problem. So, we propose ADQ, which is a product of capacity and TTI, for the element of the ADQ. We combine the ADQ and negative constants which indicate resource overlap and QoS constraint as the reward of the agent. Through simulation, we could see that the agent maximizes sum ADQ without violating QoS constraints. We can also check whether the QoS constraint affects the resource selection rate. Finally, the proposed method shows superior performance than the benchmarks, in the point that it can simultaneously maximize ADQ and guarantee QoS constraint.

REFERENCES

[1] K. Abboud, H. A. Omar, and W. Zhuang, "Interworking of DSRC and cellular network technologies for V2X communications: A survey," *IEEE Trans. Veh. Technol.*, vol. 65, no. 12, pp. 9457–9470, Dec. 2016.

[2] O. Kaiwartya, A. H. Abdullah, Y. Cao, A. Altameem, M. Prasad, C.-T. Lin, and X. Liu, "Internet of Vehicles: Motivation, layered architecture, network model, challenges, and future aspects," *IEEE Access*, vol. 4, pp. 5356–5373, 2016.

[3] J. Contreras-Castillo, S. Zeadally, and J. A. Guerrero-Ibañez, "Internet of Vehicles: Architecture, protocols, and security," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3701–3709, Oct. 2018.

[4] F. Tang, Y. Kawamoto, N. Kato, and J. Liu, "Future intelligent and secure vehicular network toward 6G: Machine-learning approaches," *Proc. IEEE*, vol. 108, no. 2, pp. 292–307, Feb. 2020.

[5] J. B. Kenney, "Dedicated short-range communications (DSRC) standards in the United States," *Proc. IEEE*, vol. 99, no. 7, pp. 1162–1182, Jul. 2011.

[6] M. J. Farooq, H. ElSawy, and M.-S. Alouini, "A stochastic geometry model for multi-hop highway vehicular communication," *IEEE Trans. Wireless Commun.*, vol. 15, no. 3, pp. 2276–2291, Mar. 2016.

[7] J. Gao, M. Li, L. Zhao, and X. Shen, "Contention intensity based distributed coordination for V2V safety message broadcast," *IEEE Trans. Veh. Technol.*, vol. 67, no. 12, pp. 12288–12301, Dec. 2018.

[8] F. Lyu, H. Zhu, N. Cheng, H. Zhou, W. Xu, M. Li, and X. Shen, "Characterizing urban vehicle-to-vehicle communications for reliable safety applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 6, pp. 2586–2602, Jun. 2020.

[9] R. Ali, R. Liu, A. Nayyar, I. Waris, L. Li, and M. A. Shah, "Intelligent driver model-based vehicular ad hoc network communication in real-time using 5G new radio wireless networks," *IEEE Access*, vol. 11, pp. 4956–4971, 2023.

[10] G. Chai, W. Wu, Q. Yang, R. Liu, and F. R. Yu, "Learning-based resource allocation for ultra-reliable V2X networks with partial CSI," *IEEE Trans. Commun.*, vol. 70, no. 10, pp. 6532–6546, Oct. 2022.

[11] *NR; User Equipment (UE) Radio Transmission and Reception; Part 1: Range 1 Standalone*, 3GPP TS 38.101-1, Version 16.5.0, Nov. 2020.

[12] A. A. Zaidi, R. Baldemair, H. Tullberg, H. Bjorkegren, L. Sundstrom, J. Medbo, C. Kilinc, and I. Da Silva, "Waveform and numerology to support 5G services and requirements," *IEEE Commun. Mag.*, vol. 54, no. 11, pp. 90–98, Nov. 2016.

[13] P. Guan, D. Wu, T. Tian, J. Zhou, X. Zhang, L. Gu, A. Benjebbour, M. Iwabuchi, and Y. Kishiyama, "5G field trials: OFDM-based waveforms and mixed numerologies," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 6, pp. 1234–1243, Jun. 2017.

[14] L. Zhang, A. Ijaz, P. Xiao, A. Quddus, and R. Tafazolli, "Subband filtered multi-carrier systems for multi-service wireless communications," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1893–1907, Mar. 2017.

[15] B. Yang, L. Zhang, O. Onireti, P. Xiao, M. A. Imran, and R. Tafazolli, "Mixed-numerology signals transmission and interference cancellation for radio access network slicing," *IEEE Trans. Wireless Commun.*, vol. 19, no. 8, pp. 5132–5147, Aug. 2020.

[16] Y. Ben-Shimol, I. Kitroser, and Y. Dinitz, "Two-dimensional mapping for wireless OFDMA systems," *IEEE Trans. Broadcast.*, vol. 52, no. 3, pp. 388–396, Sep. 2006.

[17] T. Wang, H. Feng, and B. Hu, "Two-dimensional resource allocation for OFDMA system," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, May 2008, pp. 1–5.

[18] L. You, Q. Liao, N. Pappas, and D. Yuan, "Resource optimization with flexible numerology and frame structure for heterogeneous services," *IEEE Commun. Lett.*, vol. 22, no. 12, pp. 2579–2582, Dec. 2018.

[19] W. Sui, X. Chen, S. Zhang, Z. Jiang, and S. Xu, "Energy-efficient resource allocation with flexible frame structure for heterogeneous services," in *Proc. Int. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom) IEEE Cyber, Phys. Social Comput. (CPSCom) IEEE Smart Data (SmartData)*, Jul. 2019, pp. 749–755.

[20] Q. Mao, F. Hu, and Q. Hao, "Deep learning for intelligent wireless networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2595–2621, 4th Quart., 2018.

[21] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y.-C. Liang, and D. I. Kim, "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3133–3174, 4th Quart., 2019.

[22] H. Ye, G. Y. Li, and B. F. Juang, "Deep reinforcement learning based resource allocation for V2V communications," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3163–3173, Apr. 2019.

[23] Y. Wei, F. R. Yu, M. Song, and Z. Han, "User scheduling and resource allocation in HetNets with hybrid energy supply: An actor-critic reinforcement learning approach," *IEEE Trans. Wireless Commun.*, vol. 17, no. 1, pp. 680–692, Jan. 2018.

[24] M. Zambianco and G. Verticale, "Spectrum allocation for network slices with inter-numerology interference using deep reinforcement learning," in *Proc. IEEE 31st Annu. Int. Symp. Pers., Indoor Mobile Radio Commun.*, Aug. 2020, pp. 1–7.

[25] Z. Kaleem, A. Ahmad, O. Chughtai, and J. J. P. C. Rodrigues, "Enhanced max-min rate of users in UAV-assisted emergency networks using reinforcement learning," *IEEE Netw. Lett.*, vol. 4, no. 3, pp. 104–107, Sep. 2022.

[26] A. D. Mafuta, B. T. J. Maharaj, and A. S. Alfa, "Decentralized resource allocation-based multiagent deep learning in vehicular network," *IEEE Syst. J.*, vol. 17, no. 1, pp. 87–98, Mar. 2023.

[27] D. Han and J. So, "Energy-efficient resource allocation based on deep Q-network in V2V communications," *Sensors*, vol. 23, no. 3, p. 1295, Jan. 2023.

[28] S. S. Tripathy, A. L. Imoize, M. Rath, N. Tripathy, S. Bebortta, C.-C. Lee, T.-Y. Chen, S. Ojo, J. Isabona, and S. K. Pani, "A novel edge-computing-based framework for an intelligent smart healthcare system in smart cities," *Sustainability*, vol. 15, no. 1, p. 735, Dec. 2022.
[29] S. Bebortta, S. S. Tripathy, S. Basheer, and C. L. Chowdhary, "DeepMist: Toward deep learning assisted mist computing framework for managing healthcare big data," *IEEE Access*, vol. 11, pp. 42485–42496, 2023.
[30] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.
[31] R. Bellman, "Dynamic programming," *Science*, vol. 153, no. 3731, pp. 34–37, Jul. 1966.
[32] *5G; Study on Channel Model for Frequencies From 0.5 to 100 GHz*, 3GPP TR 38.901, Version 17.0.0, Apr. 2022.
[33] 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, no. 7540, pp. 529–533, 2015.
[34] *Technical Specification Group Radio Access Network: Study LTE-Based V2X Services*, document 3GPP TR 36.885, 3rd Generation Partnership Project, Release 14, Jun. 2016.
[35] *NR; Physical Channels and Modulation*, document 3GPP TS 38.211, Version 17.2.0, Jul. 2022.
[36] A. Manzoor, S. M. A. Kazmi, S. R. Pandey, and C. S. Hong, "Contract-based scheduling of URLLC packets in incumbent EMBB traffic," *IEEE Access*, vol. 8, pp. 167516–167526, 2020.

HEETAE JIN received the B.S. degree from the Department of Electronics Engineering, Pusan National University, Busan, South Korea, in 2016, where he is currently pursuing the Ph.D. degree in electronics engineering. His research interests include the V2V, V2X communications, and applied machine learning in communications.

JEONGBIN SEO received the B.S. degree from the Department of Electronics Engineering, Pusan National University, Busan, South Korea, in 2019, where he is currently pursuing the Ph.D. degree in electronics engineering. His research interests include the index modulation and applying machine learning in communication system.

JEONGHUN PARK received the B.S. degree from the Department of Electronics Engineering, Pusan National University, Busan, South Korea, in 2021, where he is currently pursuing the master's degree in electronics engineering. His research interest includes applied machine learning for communications.

SUK CHAN KIM (Senior Member, IEEE) received the B.S.E. degree (summa cum laude) in electronics engineering from Pusan National University (PNU), Busan, South Korea, in February 1993, and the M.S.E. and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in February 1995 and 2000, respectively. He has been a Professor with the Department of Electronics Engineering, PNU, since 2002. He is a member of the Institute of Electronics Engineers of Korea (IEEK) and the Korean Institute of Communication Sciences (KICS), and a Senior Member of the Institute of Electrical and Electronics Engineers (IEEE). His research interests include mobile communications, statistical signal processing, and applied machine learning for healthcare and communications. He won the Haedong Paper Award from KICS, in 2005.