Multi-Agent Reinforcement Learning based Adaptive Parameter Optimization for Semi-Persistent Scheduling in C-V2X Mode 4

1st Pengcheng Fan
School of Software Technology
Dalian University of Technology
Dalian, China
fanpc@mail.dlut.edu.cn

2nd Xinyu Chen
School of Software Technology
Dalian University of Technology
Dalian, China
zhangjiaming@mail.dlut.edu.cn

3rd Jing Zhao*
School of Software Technology
Dalian University of Technology
Dalian, China
zhaoj9988@dlut.edu.cn

4th Ning Lu
School of Software Technology
Dalian University of Technology
Dalian, China
luning0408@mail.dlut.edu.cn

5th Ping Wang
School of Software Technology
Dalian University of Technology
Dalian, China
2019wangping@mail.dlut.edu.cn

***Abstract*—The Third Generation Partnership Project (3GPP) has standardized cellular vehicle-to-everything (C-V2X) Mode 4 to facilitate direct communication between intelligent connected vehicles. In Mode 4, vehicles autonomously reserve and select wireless spectrum resources through sensing-based semi-persistent scheduling (SPS). However, the half-duplex transmissions and hidden terminal problems in SPS could degrade the quality of services (QoS), possibly making it hard to meet the requirements for basic safety services. To support the SPS algorithm's performance in highly congested conditions, this paper presents QMIX-SPS, an adaptive parameter optimization methodology. It also proposes a new performance metric, the signal-to-interference-plus-noise ratio (SINR) achievement rate, as a means of evaluating communication network effectiveness. We developed a multi-agent vehicular communication framework in which parameters of the SPS, including transmission power, resource reservation probabilities, resource reservation counter-offset steps, and candidate resource ratios, are periodically adjusted under the guidance of the QMIX reinforcement learning algorithm. Our proposed resource allocation algorithm utilizes the mechanism of value decomposition to solve the reward allocation problem when competition and collaboration coexist in a V2X environment. Simulation results show that QMIX-SPS outperforms the baseline algorithm. Furthermore, the algorithm has excellent stability and flexibility and maintains excellent compatibility with the SPS.**

***Index Terms*—C-V2X Mode 4, spectrum resource allocation, semi-persistent scheduling, multi-agent reinforcement learning**

I. Introduction

The Internet of Vehicles (IoV) has become an essential component in enhancing road traffic systems' safety, efficiency, and convenience due to the development of Intelligent Transport Systems (ITS). In recent years, the concept of vehicle-to-everything (V2X) communications has gained the interest of both business and academics, emerging as a key component of vehicular technology. Two main technologies facilitate vehicle network communication: Cellular Vehicle-to-Everything (C-V2X) and Dedicated Short-Range Communication (DSRC) [1]. While C-V2X was introduced by the Third Generation Partnership Project (3GPP) in its Release 14 [2], DSRC is based on the IEEE 802.11p standard. Comparing C-V2X to DSRC, greater scalability, higher Quality of Service (QoS), and a larger communication coverage range are provided by utilizing LTE and 5G technology. Mode 3 and Mode 4 are the two radio resource allocations that C-V2X provides for vehicle-to-vehicle (V2V) direct communication, as seen in Fig. 1. While Mode 4 enables autonomous resource selection by vehicles via the PC5 interface, distributing radio resources according to sensing-based semi-persistent scheduling (SPS), Mode 3 entails centralized resource scheduling through the

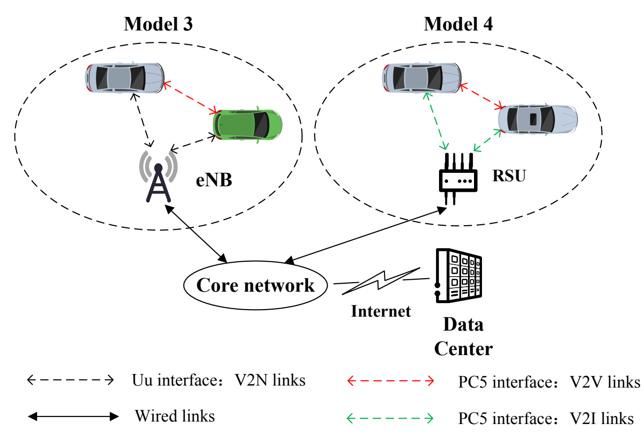


Fig. 1: Communication modes of C-V2X.

base station's UU interface. The paper will concentrate on the study of the SPS for vehicle-to-vehicle (V2V) communication within C-V2X Mode 4.

The performance of the V2X resource allocation algorithms is extremely important for achieving reliable direct communication and broadcasting safety messages between adjacent vehicles. In previous work [3–7], people have studied the performance of the SPS algorithm through analytical models and simulation models and found that appropriately adjusting SPS parameters can provide better performance. Dayal et al. [8] proposed an adaptive SPS scheme to adjust the resource reservation interval under different vehicle traffic scenarios. This reduces interference between adjacent vehicles and increases the effective communication distance. Based on the analytical model, Gu et al. [9] optimized parameters such as the SPS resource reservation interval to reduce the congestion level of the channel. Although these methods have optimized SPS parameters to a certain extent, there are still problems with half-duplex transmission and hidden terminals, so some research efforts have turned to improving the SPS process. Kim et al.[10] proposed a method based on intelligent part sensing that enhances the sensing capability of SPS by minimizing the number of blind decodes. Wang et al. [11] proposed an in-platoon collaborative sensing method to solve the hidden terminal problem in the platoon scenario. Gu et al.[12] used reinforcement learning methods to optimize the random selection process of resource blocks. With the introduction of more advanced V2X applications, traditional optimization schemes cannot meet the diverse performance requirements of resource allocation algorithms, and some studies have shown great potential by introducing reinforcement learning-based resource allocation schemes through Markov modeling. Liang et al. [13] innovatively introduced a multi-intelligent resource allocation method to solve the different QoS requirements of V2V and V2I links. Hegde et al. [14] considered influencing factors such as V2X data traffic pattern, vehicle traffic density, and radio resource utilization and proposed an AI-based resource scheduler to improve the comprehensive performance of the V2X resource allocation algorithm. Parvini et al. [15] proposed a task reward decomposition mechanism to further improve the performance of the resource allocation algorithm based on deep reinforcement learning (DRL) in the formation scenario.

Based on the research of related work, this paper proposes a DRL-based SPS algorithm that integrates the value decomposition network called QMIX-SPS [16]. We provide a novel metric called the Signal-to-Interference-plus-Noise Ratio (SINR) achievement rate to carry out a comprehensive optimization of the network performance.QMIX-SPS can guide vehicles to select the parameters of SPS periodically to adapt to the rapidly changing traffic conditions and network topology. We use an architecture combining distributed execution and centralized training to handle the value assignment problem in a multi-agent environment. The main contributions of our work can be summarized in three aspects:

- For V2V links to meet the reliability and timeliness requirements necessary to provide essential security services for the Internet of Vehicles (IoV), this work develops a utility index, or SINR achievement rate, based on transmission delay and packet transmission interruption rate. The utility index considers the impact of delay and stability on service quality in addition to the probability of successful transmission.
- This paper presents a novel attempt at simultaneously optimizing SPS parameters through reinforcement learning techniques. In the case of intense channel competition, the establishment of effective communication and cooperation between vehicles can successfully mitigate resource conflicts and improve the overall communication performance of the network.QMIX-SPS facilitates efficient resource allocation by enabling implicit collaborative communication between vehicles. The simulation results demonstrate superior performance compared to baseline algorithms such as SPS.
- This paper is the initial study that investigates the compatibility of C-V2X resource allocation. Because of the continuous evolution of C-V2X technology, multiple resource allocation algorithms will inevitably coexist in the same environment. We fully considered the impact of introducing QMIX-SPS in the SPS environment. The simulation results show that the QMIX-SPS we proposed has almost no impact on the original SPS environment and also improves the stability and reliability of the overall network.

The subsequent sections of this paper are organized in the following manner. Section II presents our system model. Section III introduces the problem description. Section IV outlines the proposed QMIX-SPS algorithm. Section V evaluates the proposed resource allocation algorithm through simulation. Finally, Section VI provides a summary of the paper.

II. System Model

This paper assumes that in a C-V2X Mode 4 environment without base station coverage, there are v vehicles in total using the PC5 interface for direct communication between vehicles. As shown in Fig. 2, in C-V2X Mode 4, vehicles sense the idle status of resources by monitoring the historical usage of channel resource blocks. Through the reference signal received power (RSRP) information recorded in the sensing window size fixed to 1000 ms, the average RSRP value of multiple periods is obtained. When the average value is higher than the threshold or there is SCI information indicating the resource to be utilized, the resource is considered busy; otherwise, it is considered idle. Then the filtered idle resources form the available resource list L_1, the number of which accounts for at least 20% of the total number of candidate resources. If the quantity is insufficient, gradually increase the RSRP threshold by 3 dBm until the quantity requirement of L_1 is met. When selecting resources, the selected time domain range is $[t + T_1, t + T_2]$, where t is the current time, T_1 represents the system processing delay ($T_1 \leq 4$ ms), and T_2 is the packet delay budget required by the application (20ms$\leq T_2 \leq$100ms). The resources in the selected time domain are called candidate single-subframe resources (CSRs). The Received Signal Strength Indication (RSSI) is used to select a group of resources with the lowest RSSI from L_1, namely L_2, to eliminate RSRP measurement errors. Moreover, the number of L_2 must be equal to 20% of the total number of optional CSRs. Finally, the vehicle transmits data according to the specified resource reservation interval (RRI). The RRI determines the value range of the Resource Reservation Counter (RC). RRI is greater than or equal to 100ms, equal to 50ms, and equal to 20ms, and RC is between [5,15], [10,30], and [25,75] respectively. After each data transmission, RC is decreased by 1 until RC is 0. When RC is 0, it is determined whether to retain or reselect resources based on the Resource Keep probability. If reselection is determined, a resource is randomly selected from L_2. However, regardless of whether resources are reselected, RC needs to be reset.

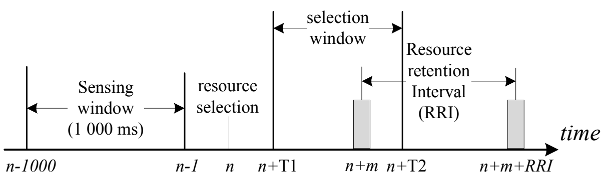


Fig. 2: Procedure of sensing-based semi-persistent scheduling.

Orthogonal Frequency Division Multiplexing (OFDM) technology is used in the Internet of Vehicles. Its principle is to convert selected channels in the frequency domain into parallel flat channels on multiple subcarriers. Several consecutive subcarriers are divided into a spectrum subchannel, assuming that the channel fading is approximately the same within a subchannel and is independent of different subchannels. The available spectrum bandwidth of V2V direct communication is w MHz, which is divided into k subchannels, and it is assumed that different subchannels do not interfere with each other. From the time domain, every 100 ms is a broadcast period. Within the same broadcast period, each vehicle can only reserve one resource block at the same time. The vehicle needs to use the adaptive strategy π to select spectrum resources from the spectrum resource pool to transmit periodic broadcast messages to surrounding vehicles. The transmission power of vehicle i in the tth period is P_t^i, and the selected spectrum resource is rs_t^i. The transmission gain between vehicles changes dynamically as the vehicle moves. Since the single period time is short, assuming the transmission gain is the same in a single period, the transmission gain of vehicle i and vehicle j in the tth period is:

$$g_t^{i,j} = PL_t^{i,j} + SF \quad (i \neq j) \tag{1}$$

$PL_t^{i,j}$ and SF are path loss and shadow fading respectively. vehicle j when receiving the signal from car i, the interference from other vehicles is described as:

$$I_t^{(j,i)} = \sum_{i^* \neq j, i^* \neq i} B_t^{(j,i^*)} P_t^{i^*} g_t'^{(j,i^*)} \tag{2}$$

where $B_t^{(j,i^*)}$ is 1 if and only if rs_t^i and $rs_t^{i^*}$ are the same, otherwise it is 0. Then the description of SINR is:

$$\gamma_t^{(j,i)} = \frac{P_t^i g_t'^{(i,j)}}{\sigma^2 + I_t^{(j,i)}} \tag{3}$$

where the numerator term represents the effective received power of vehicle j to vehicle i, and σ^2 is the noise power. According to Shannon's second theorem, the transmission rate of vehicle j receiving vehicle i is $T_t^{(j,i)}$, which is given by:

$$T_t^{(j,i)} = \frac{w}{k} log_2(1 + \gamma_t^{(j,i)}) \tag{4}$$

III. Problem Formulation

Various application services in C-V2X have distinct QoS requirements. The V2V direct link is primarily responsible for transmitting safety application messages, such as the CAM information that the vehicle periodically broadcasts, and contains the most important basic vehicle data, including position, speed, direction, and other data. Such messages have a great impact on the security of intelligent network-connected vehicles. This link needs to ensure the stability and timeliness of data packet transmission. We choose to use the packet transmission interruption rate to measure the stability of data packet delivery and the sending delay to measure the timeliness.

Based on [13], it can be obtained that under the constraint of the upper limit of transmission interruption probability p_o, the SINR of data packet transmission needs to satisfy the following formula:

$$\gamma_t^{(j,i)} \geq \frac{\gamma_{min}}{ln\left(\frac{1}{1-p_o}\right)} \tag{5}$$

where $\gamma_m in$ represents the SINR threshold transmission delay, which is defined as follows:

$$\frac{Z}{T_t^{(j,i)}} \leq td_{max} \tag{6}$$

In (6), Z represents the data packet size, and td_{max} represents the upper limit of the sending delay required by the security application. Combining (5) and (6), we can simplify the service quality requirements into constraints on SINR:

$$\gamma_t^{(j,i)} \geq \max\left(\frac{\gamma_{min}}{ln\left(\frac{1}{1-p_o}\right)}, 2^{\frac{Z \cdot k}{td_{max} \cdot w}} - 1\right) \tag{7}$$

To maximize $\gamma_t^{((j,i)}$ for a single vehicle, one can enhance the transmission power P_t^i. However, this will cause interference with other vehicles that select the same spectrum resources, causing the SINR of these vehicles to decrease. A strategy that is better for some vehicles may not necessarily be better overall. Therefore, the understanding of this problem should be considered from a global perspective. This paper aims to make as much data transmission as possible satisfy the constraints of (7). We design the global SINR achievement rate as an evaluation index to measure strategy performance:

$$PAS = \frac{N_a}{N_{total}} \tag{8}$$

where N_a represents the number of packet transmissions whose SINR satisfies the constraint condition of (7), and N_total represents the total number of transmissions. This paper hopes to maximize PAS by dynamically and adaptively adjusting the parameters and transmit power of the SPS algorithm. Therefore, we have the following formal description of this problem:

$$\begin{gathered} max \text{ PAS} \\ s.t. \begin{cases} 0 \leq H \leq 1 \\ 0 \leq F \leq 1 \\ C \in \{0,1,2\} \\ 0 \leq P \leq P_{max} \end{cases} \end{gathered} \tag{9}$$

where H is the resource reservation probability, F is the candidate resource filtering coefficient, C is the reselection counter step offset, and P is the transmission power.

IV. RL Based Resource Selection Algorithm

This section mainly introduces the modeling of multi-agent environments and the enhanced SPS algorithm based on the QMIX algorithm.

A. Modeling of Multi-Agent Environments

Reinforcement learning is a machine learning technique that involves agents learning optimal strategies by interacting with the environment through interactive trial and error. When formalizing a reinforcement learning problem, there are a few key elements that define the interaction between the agent and the environment. For this problem, vehicles need to make decisions independently based on observations of the environment. According to the decentralized partially observable Markov decision process (Dec-POMDP) [17], the reinforcement learning elements can be defined as $< S, A, O, R, \tau, \delta >$. Among them, S represents the state of the environment, O represents the agent's observation value of the environment state, A represents the action space in which the vehicle makes decisions, R represents the reward obtained by the vehicle's decision, δ represents the discount factor of the reward, and the multi-agent reinforcement learning The objective of the optimization is to maximize the cumulative vehicle rewards:

$$\max_{\pi_i} \mathcal{J}_i(\pi_i) \tag{10}$$

Where $\mathcal{J}_i(\pi_i) = \mathbb{E}[\sum_{t=0}^{\infty} \delta^t R_{t+1}^i | s_0^i]$, π_j represents the strategy of vehicle i. Each vehicle interacts with the vehicular network environment as an agent and takes actions based on state observations, aiming to solve the optimization problem (9), or, in other words, maximize its total expected return (10). At each time t, the vehicle takes a decision a_t based on the observed environment state o_t. The environment will be transferred to s_{t+1}, and then the vehicle will receive a reward r based on the decision made at the previous time. In the multi-agent environment architecture we established, the state observation space O, action space A, state space S, and reward function R are defined as follows:

1) Observation: In the actual environment under C-V2X Mode 4, the information that the vehicle can stably observe is very limited, mainly including two aspects: vehicle driving information and spectrum resource sensing information. This paper uses the following tuple to describe the observation information of the tth time step after vehicle i performs the action a_{t-1} at the tth time step:

$$o_t^i = \{Speed_t^i, Vec_t^i, \mathbf{CS}_t^i, RC_t^i\} \tag{11}$$

Vehicle driving information includes two elements: $Speed$ and Vec, which are vehicle speed and vehicle driving direction respectively. These two pieces of information are instrument information that can be stably obtained when driving in the actual environment. Spectrum resource sensing information $\mathbf{CS}_t^i$: includes five elements: $CS1_t^i$, $CS2_t^t$, $CS3_t^t$,$CS4_t^t$, $CS5_t^t$. These five elements are the five statistics of the RSSI of the spectrum resources by the vehicle in the past broadcast period. After excluding the interference of half-duplex, these five statistics respectively represent the total number of resources, the number of resources with RSSI greater than the threshold, and the resource RSSI Cumulative sum, resource RSSI mean, and resource RSSI standard deviation. It is used to help the model perceive the overall situation of spectrum resources. In addition to the above observations of the environment, there is also a description of the protocol status. This paper uses the resource reselection counter RC_t^i to help the model understand the current status of the SPS protocol.

2) Action: :The action performed by vehicle i at time step t is defined as follows:

$$a_t^i = (H_t^i, F_t^i, C_t^i, P_t^i) \tag{12}$$

where H_t^i represents the dynamic reservation probability of spectrum resources. When the resource reservation counter returns to 0, whether to reserve resources will be decided based on the resource reservation probability. This paper uses this parameter to control the stability of resource reservations. F_t^i represents the proportion of candidate resources. Before resource reselection, it is necessary to filter the candidate resource set with the lowest RSSI from the spectrum resource pool according to the proportion of F_t^i, and then randomly select tile resources from the set. C_t^i is the offset step size of the resource reservation counter. In the SPS protocol, every time the vehicle performs a periodic broadcast, the resource reservation counter will be decremented by 1. When it returns to 0, resource reselection is possible. we use C_t^i to adjust the single change step size of the resource reservation technology. That is, each time a periodic broadcast is performed, the resource reservation counter is decremented by $(1 + C_t^i)$.P_t^i represents signal transmission power.

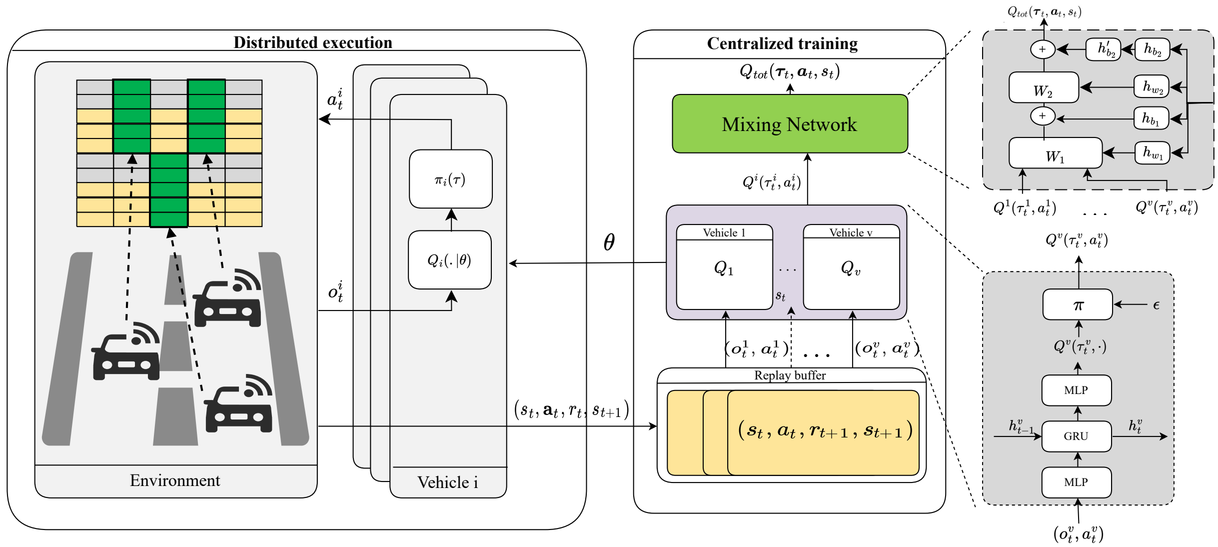


Fig. 3: QMIX-SPS resource allocation algorithm framework

3) State: Under the centralized training and distributed execution architecture, global state information is only used during the training phase to help each agent model modify its strategy. Different from the state observation space, the training phase needs to use as comprehensive state information as possible to assist the agent in training. This information includes the following aspects: observation information of each vehicle, vehicle action information, and supplementary related information. Global status information at time t:

$$s_t = \{\mathbf{O_t}, \mathbf{a_{t-1}}, Location_t\} \tag{13}$$

where $\mathbf{O_t}$ is the collection of environmental observation data of all vehicles following the execution of action a_{t-1}, and $\mathbf{a_{t-1}}$ is the collection of actions that all vehicles select to do at the $t-1$th time step. The position of every vehicle at the tth time step, or $Location_t$, provides the extra relationship information. During the training phase, this parameter is intended to help the model measure the vehicle's distribution status.

4) Reward: The evaluation of the scheduling strategy in this paper is based on the SINR achievement rate PAS. The higher the PAS, the higher the proportion of communications that meet QoS constraints. r_t represents the global reward value for the $t-1$ event step as follows:

$$r_t = PAS_t - b_{sps} \tag{14}$$

where b_{sps} is the artificially set baseline. Since PAS is always positive, if PAS is used directly as a reward, the reward will always be positive, that is, any action selected will have a positive reward, which may cause difficulties in model training. Therefore, the PAS baseline b_sps is introduced, and its value is the mean PAS value obtained by using the original SPS protocol under the same conditions. Essentially, it involves a type of comparative learning.

B. Dynamic Adaptive Parameters Optimization Algorithm QMIX-SPS

On the basis of multi-agent environment modeling, this paper proposes a joint dynamic parameters adaptive optimization algorithm for SPS based on the QMIX algorithm. The QMIX algorithm helps agents collaborate better to achieve global optimal results by constructing a hybrid network combined with the value function of a single agent. The overall architecture of the proposed algorithm is shown in Fig. 3. The proposed framework comprises two components: the agent network and the mixing network.

The agent network employs a deep recurrent Q network model to directly determine actions, integrating the recurrent unit (GRU) model within the deep recurrent neural network. This integration enables agents to leverage past trajectory data for decision-making. The agent network comprises three layers, with the GRU network situated in the middle, and the input and output layers consisting of fully connected neural networks. The input layer receives information such as the current observation data o_t^i of the vehicle and the previous action a_{t-1}^i taken. The GRU network requires the past hidden state output h_{t-1}^i of the agent as input and outputs the current hidden state h_t^i. The Q network can be represented as follows:

$$Q_i(o, a|\theta) = f_i^2(g_i(f_i^1(o_i, a_i), h_{i-1}))$$

Where θ represents the parameters of the Q network, f represents the multi-layer perceptron (MLP), and g represents the GRU network. In the output layer, the model generates the state-action value $Q(\tau_t^i, \cdot)$ for all actions based on the historical information τ_t^i, and employs a greedy strategy for action selection, as depicted in the subsequent equation:

$$\pi(\tau) = arg\max_a Q_\pi(o, a) = a_t \tag{15}$$

The final value output by the agent network is expressed as $Q^i(\tau_t^i, a_t^i)$.

The mixing network is employed for value mixing, linking the global joint action value $Q_{tot}(\mathbf{o}, \mathbf{a})$ with the action value of each agent $Q^i(\tau_t^i, a_t^i)$. The mixing network, a basic two-layer feedforward neural network, is utilized to combine the action values of individual agents in a monotonic manner to generate $Q_{tot}(\mathbf{o}, \mathbf{a})$. To adhere to the QMIX constraint of monotonicity, the weight parameter of the hybrid network is constrained to be non-negative [16].

More specifically, the weight parameters and biases of each layer of the mixing network are produced by a distinct hypernetwork. Each layer of weights in the mixing network corresponds to a hypernetwork that takes the global state s as input and outputs the parameters of the feedforward neural network. The output is in vector form, which is then reshaped into a matching matrix based on predefined rules.

Furthermore, to enhance model stability and mitigate overestimation effects, the QMIX network integrates concepts from the traditional Deep Q Learning (DQN) algorithm, including empirical buffer pools and dual Q-networks. The overall end-to-end loss function for the QMIX network can be expressed by the following equation:

$$\begin{cases} \mathcal{L}(\theta) = \sum_{i=1}^{b} \left[\left(y_{tot}^i - Q_{tot}(\boldsymbol{\tau}, \mathbf{a}, s; \theta) \right)^2 \right] \\ y_{tot} = r + \delta \max_{a'} Q_{tot}(\boldsymbol{\tau'}, \mathbf{a}', s'; \theta^-) \end{cases} \tag{16}$$

where b is the batch size sampled from the experience replay buffer for each training, and y_{tot} and θ^- represent the values obtained from the target network in the DQN as well as the network parameters.

The execution phase of the algorithm will be divided into several phases: initialization, interactive memory, and training. The main process is shown in Algorithm 1.

Algorithm 1: Training Algorithm

Input: learning rate α, replay buffer D, step limit $step_{max}$, episode limit $episode_{max}$, parameter synchronization interval $step_{inr}$, batch-size, reward factor δ

Output: θ, the parameters of mixing network, agent networks and hypernetwork

Initialise θ
$step = 0$, $\theta^- = \theta$
while $step < step_{max}$ **do**
  $t = 0$
  Get initial state s_0
  while $s_t \neq terminal$ and $t \leq episode_{max}$ **do**
    foreach i *in Vehicles* **do**
      Get available actions A_t^i for vehicle i
      $\tau_t^i = \tau_t^i \cup \{(o_t^i, a_t^i)\}$
      ϵ=epsilon-annealing($step$)
      $a_t^i = \begin{cases} \underset{a_t^i \in A_t^i}{argmax}\, Q(\tau_t^i, a_t^i) \text{ with probability } 1-\epsilon \\ \text{Randomly select action from } A_t^i \text{ with probability } \epsilon \end{cases}$
    end
    Get reward r_t and next state s_{t+1}
    D=D$\cup\{(s_t, \mathbf{a}_t, r_t, s_{t+1})\}$
    $t = t + 1$,$step = step + 1$
  end
  if $D >$ batch-size **then**
    train-batch $b \leftarrow$ random batch of episodes from D
    foreach b *in* train-batch **do**
      Calculate Q_{tot} using Mixing-network with $Hypernetwork(s; \theta)$)
      Calculate target Q_{tot} using Mixing-network with $Hypernetwork(s'; \theta^-)$)
    end
    $y_{tot} = r + \delta max Q_{tot}(\tau', a', s'; \theta^-)$
    $\Delta Q_{tot} = y_{tot} - Q_{tot}$
    $\Delta\theta = \nabla_\theta(\Delta Q_{tot})^2$
    $\theta = \theta - \alpha\Delta\theta$
  end
  Synchronize parameters every $step_{inr}$: $\theta^- \leftarrow \theta$
end

During the initialization phase of the model, various parameters such as the network parameters denoted as θ, exploration probability ϵ, learning rate α, and buffer pool D undergo initialization operations. The hypernetwork and agent network parameters in the QMIX model are randomly initialized using a Gaussian method to introduce variability in decision-making among vehicles. The exploration probability ϵ is typically initialized close to 1 to mitigate the risk of the model getting stuck in a local optimum early in the training process, with the learning rate being set similarly. Agents are encouraged to explore during action selection to enhance the fitting of the action value function. Employing an exploratory strategy aids in generating a more diverse training dataset, thereby facilitating a more accurate fitting of the action value function by the agent network.

In the interactive memory phase, the agent engages with the environment, storing observations, status, actions, and rewards pre and post-interaction in the experience replay buffer. During interactions with the environment, the agent can choose actions from the available action space using either an exploration or greedy strategy. This mechanism ensures that the training of the agent network (action value function) is not influenced by temporal locality, thereby enhancing model stability. The size of the experience replay pool in this study is ten times the size of the action space.

Throughout the training phase, the model extracts a batch of interaction data from the experience replay pool for batch training, enhancing training efficiency. The learning rate decreases progressively during training to strike a balance between training speed and the risk of falling into a local optimum. At fixed intervals, all parameters of the agent network are copied to the target network. The calculation of y_t from the target network and reward value, as (16), guides the backpropagation of the loss value gradient to optimize each network parameter of the model.

V. Simulation Results and Analysis

In this section, we introduce the simulation tools used in the study and the relevant experimental parameters. Then, the training process analysis, performance evaluation, and compatibility analysis of the proposed reinforcement learning algorithm are performed respectively.

A. Simulation Setup

Our simulation experiments are based on the open-source Python simulator Simulators-for-SPS [9]. We made some modifications to the simulator to make it more compliant with the TR 36.885 [18], mainly changing the channel model to Winner+B1 required by 3GPP. At the same time, the simulation experiments refer to the 3GPP C-V2X simulation guide [19]. We build an urban scenario with a 1299m×750m Manhattan grid. The scenario is composed of 3×3 units, and each road is two-way and four-lane. Vehicles move smoothly in the urban grid. We considered different vehicle densities and vehicle kinematics in the simulation and generated real traffic trajectory data by the road traffic simulator Simulation of Urban MObility (SUMO). The parameters of the simulation experiments are shown in Table I below. When training our model, based on our experience and multiple experiments, the learning rate of the Q network is set to 0.00007 and the batch size of the sampled data is 256. More detailed parameters for model training are shown in Table II.

TABLE I: Simulation parameters

Parameter	Value
Vehicle speed limit	60 km/h
Vehicle average initial speed	36 km/h
Vehicle acceleration	[-4.5 m/s^2,4.5 m/s^2]
Carrier frequency	5.9 GHz
Channel bandwidth	10 MHz
Subchannels per subframe	5
RBs per subchannel	10
Modulation and coding scheme	MCS 4
Path loss model	WINNER+B1
Antenna gain	3 dB
Antenna height	1.5 m
Shadow fading standard and deviation	3 dB, 4 dB
RSRP threshold	-128 dBm
Resource retention period	100ms
Message sending frequency	10HZ
Noise spectral density	-174 dBm/HZ
Packet size	190 Bytes
perceived distance	150 m
resource awareness period	1000 ms
Resource selection window size	100 ms
Resource selection window	T_1=1, T_2=100
Resource counter selection range	[5,15]

B. Evaluation Metrics

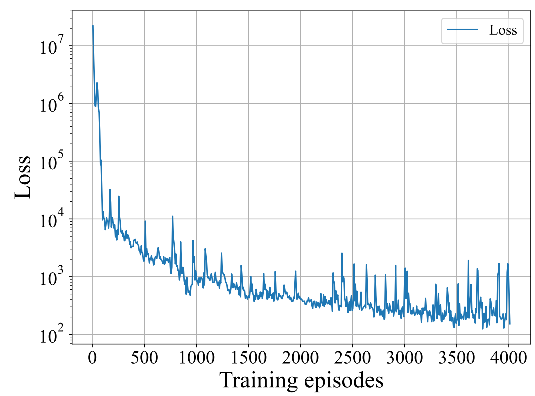
To evaluate the performance of different resource selection algorithms, we introduce the following metrics in addition to PAS.

TABLE II: Model training parameters

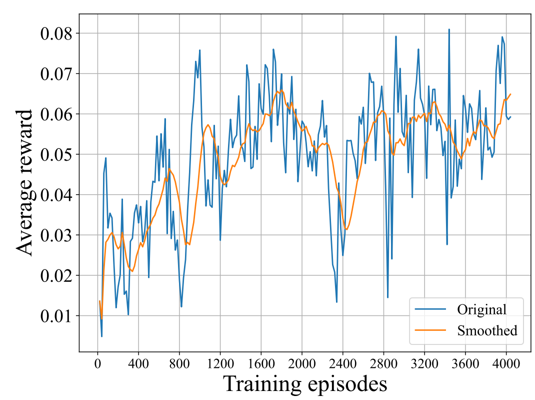
Parameter	Value
learning rate α	0.00007
Q network synchronization episodes interval	100
Experience replay buffer memory size	50000
Discount rate δ	0.99
Number of steps per training	34
Total training episode	4000
Mini-batch size b	256
Exploitation initial value	1.0
Exploitation probability minimum	0.05
Exploitation decreasing step	2.375×10^{-6}

- **Packet delivery rate (PDR)**: The ratio of the number of packets actually successfully received by the vehicle to the number of packets expected to be received by the vehicle.
- **Transmission speed**:The transmission rate can be calculated from (4). This metric can be used to measure the instantaneous capacity of the link.

C. Simulation Result



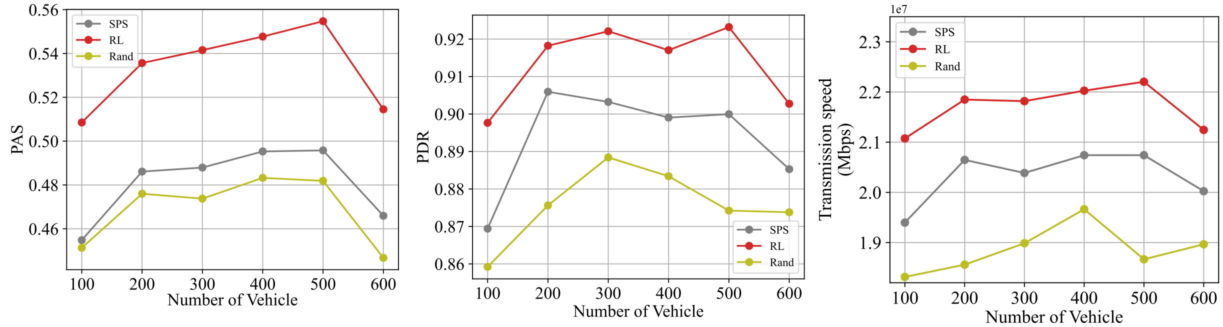
(a) Training loss



(b) Average reward per episode

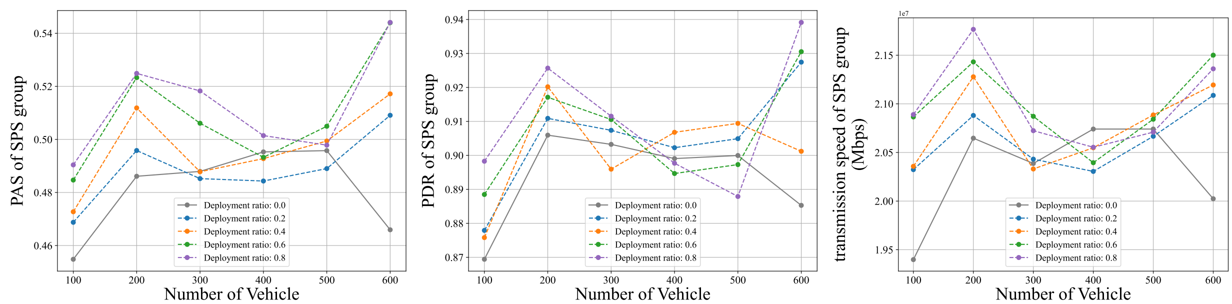
Fig. 4: Training performance evaluation for QMIX-SPS

1) Training process analysis: Figure. 4(a) shows the training performance of our proposed QMIX-SPS algorithm with



(a) Performance of PAS (b) Performance of PDR (c) Performance of Transmission speed

Fig. 5: Performance of QMIX-SPS.



(a) Compatibility of PAS (b) Compatibility of PDR (c) Compatibility of Transmission speed

Fig. 6: Compatibility of QMIX-SPS.

a mini-batch size of 256. It is observed that the loss function value of the mixing network decreases rapidly with the increase of training episodes until it converges to a minimal value. Figure. 4(b) shows the changing trend of the average reward as the number of training sets increases. The average reward climbs oscillate over time before reaching a dynamic equilibrium, which is a normal phenomenon in reinforcement learning. It can be seen that the training algorithm we proposed performs well at convergence.

2) Performance of the Algorithm: We investigate the performance of the SPS algorithm under varying vehicle densities in an urban scenario. When the total vehicle count is less than or equal to 500, the SPS algorithm's performance across metrics is not significantly affected by the vehicle count. However, once the vehicle count exceeds 500, we observe a relative decline in the SPS algorithm's performance, aligning with the limited available spectrum resources. This suggests the SPS algorithm's adaptive capabilities in low-density scenarios, but limited performance improvements in high-density settings. To address this, we propose the QMIX-SPS algorithm, which builds on the SPS mechanism with adaptive control of the algorithm parameter and signal transmit power. Experiments confirm the QMIX-SPS algorithm significantly enhances key C-V2X mode 4 communication metrics in the urban Manhattan grid scenario. Our findings highlight the need for scalable communication strategies to ensure reliable V2X performance in high-density urban environments. The QMIX-SPS algorithm offers a promising approach to improving V2X communication performance in complex urban settings.

3) Compatibility analysis: Since C-V2X is an evolving technology, different versions of MAC layer protocols must coexist. This paper considers that the newly proposed algorithm needs to maintain a certain degree of compatibility with the existing SPS protocol. The compatibility here refers to reducing the impact of introducing new resource allocation algorithms into the SPS environment in the original environment. Specifically, we consider different proportions of QMIX-SPS vehicles deployed in the SPS environment. The deployment proportion is equal to the ratio of the number of vehicles deploying QMIX-SPS to the total number of vehicles. As can be seen from Fig. 6, overall the QMIX-SPS algorithm has good compatibility with SPS, and the deployment of QMIX-SPS will not affect the performance of the SPS vehicle network. On the contrary, deploying QMIX-SPS can improve the performance of the SPS network. This performance improvement is particularly obvious in the case of low-density (100 vehicles) and high-density (600 vehicles), which illustrates the coordinating role of QMIX-SPS in vehicle communication networks. The performance optimization of the SPS network by QMIX-SPS is not comprehensive. For example, QMIX-SPS decreases the performance of the SPS network in an 80% deployment ratio setting with 400 vehicles.

VI. Conclusion

In this paper, we first introduce the spectrum resource allocation problem in C-V2X Mode 4 in detail and describe the system modeling and formal definition of this problem. Before formally introducing the algorithm proposed in this paper, we explain the design of reinforcement learning elements for this problem, including the design of action space, state space, observation space, and reward function. We provide detailed explanations for the considerations and purposes behind each element. Subsequently, this section introduces the proposed QMIX-SPS algorithm including the implementation process, optimization mechanism, and pseudo-code description. Finally, this section uses two groups of experiments to verify the impact of the algorithm on C-V2X Mode 4 communication performance and its compatibility with the original SPS algorithm. Experimental results prove that our proposed algorithm can effectively improve global communication performance while maintaining good compatibility with the SPS. In future work, we will further analyze the challenges of parameter optimization using reinforcement learning. Additionally, we will investigate how introducing resource allocation algorithms under non-SPS architecture impacts the complex V2X environments.

References

[1] A. R. Khan, M. F. Jamlos, N. Osman, M. I. Ishak, F. Dzaharudin, Y. K. Yeow, and K. A. Khairi, "Dsrc technology in vehicle-to-vehicle (v2v) and vehicle-to-infrastructure (v2i) iot system for intelligent transportation system (its): A review," *Recent Trends in Mechatronics Towards Industry 4.0: Selected Articles from iM3F 2020, Malaysia*, pp. 97–106, 2022.

[2] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 36.300, 12 2016, version 14.1.0.

[3] M. Chen, R. Chai, H. Hu, W. Jiang, and L. He, "Performance evaluation of c-v2x mode 4 communications," in *2021 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2021, pp. 1–6.

[4] M. Gonzalez-Martín, M. Sepulcre, R. Molina-Masegosa, and J. Gozalvez, "Analytical models of the performance of c-v2x mode 4 vehicular communications," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1155–1166, 2018.

[5] Z. Ali, S. Lagén, L. Giupponi, and R. Rouil, "3gpp nr v2x mode 2: Overview, models and system-level evaluation," *IEEE Access*, vol. 9, pp. 89 554–89 579, 2021.

[6] X. Gu, J. Peng, L. Cai, X. Zhang, and Z. Huang, "Markov analysis of c-v2x resource reservation for vehicle platooning," in *2022 IEEE 95th Vehicular Technology Conference:(VTC2022-Spring)*. IEEE, 2022, pp. 1–5.

[7] V. Todisco, S. Bartoletti, C. Campolo, A. Molinaro, A. O. Berthet, and A. Bazzi, "Performance analysis of sidelink 5g-v2x mode 2 through an open-source simulator," *IEEE Access*, vol. 9, pp. 145 648–145 661, 2021.

[8] A. Dayal, V. K. Shah, B. Choudhury, V. Marojevic, C. Dietrich, and J. H. Reed, "Adaptive semi-persistent scheduling for enhanced on-road safety in decentralized v2x networks," in *2021 IFIP Networking Conference (IFIP Networking)*. IEEE, 2021, pp. 1–9.

[9] X. Gu, J. Peng, L. Cai, Y. Cheng, X. Zhang, W. Liu, and Z. Huang, "Performance analysis and optimization for semi-persistent scheduling in c-v2x," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 4, pp. 4628–4642, 2022.

[10] T. Kim, Y. Kim, M. Jung, and H. Son, "Intelligent partial sensing based autonomous resource allocation for nr v2x," *IEEE Internet of Things Journal*, 2023.

[11] B. Wang, J. Zheng, N. Mitton, and C. Li, "Inp-crs: an intra-platoon cooperative resource selection scheme for c-v2x networks," *IEEE Communications Letters*, 2023.

[12] B. Gu, W. Chen, M. Alazab, X. Tan, and M. Guizani, "Multiagent reinforcement learning-based semi-persistent scheduling scheme in c-v2x mode 4," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 11, pp. 12 044–12 056, 2022.

[13] L. Liang, H. Ye, and G. Y. Li, "Spectrum sharing in vehicular networks based on multi-agent reinforcement learning," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 10, pp. 2282–2292, 2019.

[14] A. Hegde, R. Song, and A. Festag, "Radio resource allocation in 5g-nr v2x: a multi-agent actor-critic based approach," *IEEE Access*, 2023.

[15] M. Parvini, M. R. Javan, N. Mokari, B. Abbasi, and E. A. Jorswieck, "Aoi-aware resource allocation for platoon-based c-v2x networks via multi-agent multi-task reinforcement learning," *IEEE Transactions on Vehicular Technology*, 2023.

[16] T. Rashid, M. Samvelyan, C. S. De Witt, G. Farquhar, J. Foerster, and S. Whiteson, "Monotonic value function factorisation for deep multi-agent reinforcement learning," *Journal of Machine Learning Research*, vol. 21, no. 178, pp. 1–51, 2020.

[17] J. Dibangoye and O. Buffet, "Learning to act in decentralized partially observable mdps," in *International Conference on Machine Learning*. PMLR, 2018, pp. 1233–1242.

[18] 3GPP, "Study on lte-based v2x services," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 36.885, 07 2016, version 14.0.0.

[19] M. H. C. Garcia, A. Molina-Galan, M. Boban, J. Gozalvez, B. Coll-Perales, T. Şahin, and A. Kousaridas, "A tutorial on 5g nr v2x communications," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1972–2026, 2021.