mathematics

MDPI

Article

A Deep Reinforcement Learning Scheme for Spectrum Sensing and Resource Allocation in ITS

Huang Wei [1], Yuyang Peng [1,*], Ming Yue [1], Jiale Long [2], Fawaz AL-Hazemi [3] and Mohammad Meraj Mirza [4]

[1] The School of Computer Science and Engineering, Macau University of Science and Technology, Macau 999078, China; 2009853gii20018@student.must.edu.mo (H.W.); 2109853wii20001@student.must.edu.mo (M.Y.)
[2] Faculty of Intelligent Manufacturing, Wuyi University, Jiangmen 529020, China; longjiale_528@126.com
[3] Department of Computer and Network Engineering, University of Jeddah, Jeddah 21959, Saudi Arabia; fmalhazemi@uj.edu.sa
[4] Department of Computer Science, College of Computers and Information Technology, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia; mmmirza@tu.edu.sa
* Correspondence: yypeng@must.edu.mo

Abstract: In recent years, the Internet of Vehicles (IoV) has been found to be of huge potential value in the promotion of the development of intelligent transportation systems (ITSs) and smart cities. However, the traditional scheme in IoV has difficulty in dealing with an uncertain environment, while reinforcement learning has the advantage of being able to deal with an uncertain environment. Spectrum resource allocation in IoV faces the uncertain environment in most cases. Therefore, this paper investigates the spectrum resource allocation problem by deep reinforcement learning after using spectrum sensing technology in the ITS, including the vehicle-to-infrastructure (V2I) link and the vehicle-to-vehicle (V2V) link. The spectrum resource allocation is modeled as a reinforcement learning-based multi-agent problem which is solved by using the soft actor critic (SAC) algorithm. Considered an agent, each V2V link interacts with the vehicle environment and makes a joint action. After that, each agent receives different observations as well as the same reward, and updates networks through the experiences from the memory. Therefore, during a certain time, each V2V link can optimize its spectrum allocation scheme to maximize the V2I capacity as well as increase the V2V payload delivery transmission rate. However, the number of SAC networks increases linearly as the number of V2V links increases, which means that the networks may have a problem in terms of convergence when there are an excessive number of V2V links. Consequently, a new algorithm, namely parameter sharing soft actor critic (PSSAC), is proposed to reduce the complexity for which the model is easier to converge. The simulation results show that both SAC and PSSAC can improve the V2I capacity and increase the V2V payload transmission success probability within a certain time. Specifically, these novel schemes have a 10 percent performance improvement compared with the existing scheme in the vehicular environment. Additionally, PSSAC has a lower complexity.

Keywords: deep reinforcement learning; vehicle to vehicle; vehicle to infrastructure; spectrum resource allocation

MSC: 94-10

Citation: Wei, H.; Peng, Y.; Yue, M.; Long, J.; AL-Hazemi, F.; Mirza, M.M. A Deep Reinforcement Learning Scheme for Spectrum Sensing and Resource Allocation in ITS. *Mathematics* **2023**, *11*, 3437. https://doi.org/10.3390/math11163437

Academic Editor: Nadir Farhi

Received: 25 June 2023
Revised: 27 July 2023
Accepted: 1 August 2023
Published: 8 August 2023

1. Introduction

With the continuous development of new-generation mobile communication technology, the Internet of Vehicles (IoV) has attracted extensive attention all over the world in recent years because of its potential value to promote the development of intelligent transportation systems (ITSs) and smart cities. Technically, the IoV connects vehicles to the mobile network to realize the full connection function in vehicle-to-Infrastructure (V2I), vehicle-to-people (V2P), vehicle-to-network (V2N) and vehicle-to-vehicle (V2V). The spectrum resource is a necessary condition to realize all the above vehicle communications.

However, with the continuous expansion of the IoV applications and the improvement of the communication requirements, the existing spectrum resources are obviously insufficient to meet all the communication requirements. Therefore, in order to ensure the communication service of the IoV with low delay and high reliability, a new spectrum resource allocation scheme is needed [1].

The studies in the IoV area initially used traditional schemes like mathematical modeling. Compared with the traditional cellular network, the uncertain network environment including high-speed moving vehicles and changing channels in the IoV has brought unprecedented challenges to the spectrum resource allocation. Concerned with the dynamic allocation of spectrum resources among high-speed vehicles in the IoV, a dynamic spectrum allocation algorithm based on channel feedback and a graph theory coloring model was proposed in [2]. The feedback matrix is defined to analyze the occupation and communication of the channel, and the vehicle node is required to judge whether the current channel is available from the master according to the parameter values returned in the channel feedback matrix. To adapt the communication environment of the IoV with the changing number of vehicles, a three-step cognitive spectrum allocation algorithm based on a clustering structure was proposed in [3]. In addition, a spectrum sharing scheme designed for the slowly changing large-scale fading channel was proposed in [4,5], which improves the throughput of the V2I link to the greatest extent and reduces the network signaling overhead. In this scheme, V2I can share the spectrum resources with V2V. Moreover, the resources can also be shared between two V2V links.

In recent years, researchers begin to use different deep learning and reinforcement learning theories to model and solve the IoV spectrum resource allocation problem in an unknown dynamic vehicular environment [6]. For example, to meet the dynamic and diverse needs of different entities, a Q-learning framework was proposed in [7] to solve the resource allocation problem in the vehicle cloud environment. In [8], based on the deep Q-network (DQN), a joint cache and computing spectrum resource scheme was proposed for the unknown number of spectrum resources. To further solve the problems of high mobility and the centralized management of spectrum resources in most vehicle environments, a hybrid spectrum multiplexing and power allocation solution for vehicular communications was proposed in [9], and a method based on convolutional neural network (CNN) was developed to achieve real-time decisions on power allocation and spectrum reuse. In [10], a distributed multi-agent spectrum resource allocation scheme was proposed using DQN theory where each V2V link is considered as an agent, and each agent periodically observes and explores the vehicular environment. Then, due to different observation results and behavior rewards, each agent learns how to reasonably select their own transmit power and spectrum independently. In [11], aiming to solve the problem of load and resource allocation in the IoV, an optimal solution based on Q-learning that can reasonably allocate load, control transmission power, plan sub-channels, and calculate spectrum resources was proposed, which effectively reduces the overhead of the system compared with other algorithms. The high-density vehicle environment brings high-dimensional action space. Therefore, in [12], a multi-agent method based on deep deterministic policy gradient (DDPG) was used to study the spectrum resource allocation of the V2I link and V2V link under the conditions of non-orthogonal multiple access technology. It maximizes the V2I link delivering rate while meeting the strict delay and reliability constraints of V2V communication. To sum up, researchers mainly use the DQN and DDPG algorithms to solve the problem of dynamic spectrum resources allocation in the IoV. However, these two algorithms have some limitations which can be summarized as follows. DQN only shows good advantages in discrete low-dimensional behavior space; although DDPG uses the actor critic (AC) method to be suitable for continuous high-dimensional behavior space, it has poor stability because it adopts deterministic behavior strategy [13]. In a word, the existing algorithms either only deal with the discrete low-dimensional environment or have a bad performance in continuous complex environment. However, the proposed method can overcome the aforementioned issues.

To deal with the continuous change in vehicle network environment, a complete corporation soft actor critic (SAC) algorithm is proposed in this paper which has bigger V2I capacity and a higher success probability of V2V payload delivery than DQN and DDPG. In addition, in order to reduce the complexity, a parameter sharing soft actor critic scheme (PSSAC) algorithm is proposed, which performs well in a vehicle communication environment with low complexity.

The rest of this paper is organized as follows. In Section 2, the system model is described in details. Section 3 introduces the SAC-based allocation scheme. Section 4 introduces the two proposed algorithms. The simulation results are provided in Section 5 and the conclusion is given in Section 6.

2. System Model

The IoV model is shown in Figure 1 which consists of several cars, a single base station, X V2I links and Y V2V links. To ensure the high quality of the V2I link communication, it is assumed that each V2I link is allocated with different orthogonal frequency spectrum subcarriers in advance to eliminate the interference between different V2I links in the IoV environment using the spectrum sensing technology. We define that each V2I link has a fixed allocation of the subcarrier, and the transmission power of all V2I links is P_{V2I}. Since V2V link and V2I link share the same spectrum resources, each V2V link needs to select the specific spectrum and transmission power to improve the communication quality of the V2V link. Therefore, this paper will focus on the design of spectrum resources and power allocation scheme for the V2V link in the IoV to maximize the total transmission rate of the V2I and V2V links.

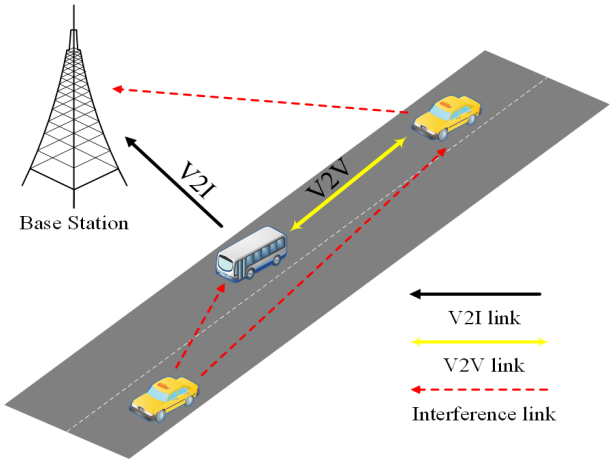


Figure 1. The model of the vehicular network.

Suppose that the channel gain of V2V link is only composed of small-scale and large-scale fading, where L_y represents the frequency-independent large-scale fading effect, namely the shadow effect and path loss, and $S_y[x]$ represents the frequency-dependent small-scale fading channel gain. Therefore, the channel power gain when y-th V2V link occupies the x-th subcarrier can be expressed as

$$G_y[x] = L_y S_y[x]. \tag{1}$$

Thus, the received signal-to-interference noise ratio corresponding to the x-th V2I link and the y-th V2V link using the x-th sub-band can be, respectively, expressed as follows

$$S_x^c[x] = \frac{P_x^c I_{x,B}[x]}{\sigma^2 + \sum_x \rho_y[x] P_y^d[x] I_{y,B}[x]} \tag{2}$$

and

$$S_y^d[x] = \frac{P_y^d[x]g_y[x]}{\sigma^2 + U_y[x]} \tag{3}$$

where σ^2 denotes the noise power, P_x^c is the x-th V2I transmit power, $P_y^d[x]$ is the y-th transmit power using the x-th sub-band, $I_{y,B}[x]$ is the interference channel between the BS and y-th V2V transmitter over the x-th sub-band, $I_{x,B}[x]$ is the interference channel between BS and the x-th V2I transmitter over the x-th sub-band, $\rho_y[x]$ denotes the binary system with $\rho_y[x] = 1$ meaning that the y-th V2V link is using the x-th sub-band and $\rho_y[x] = 0$ is vice versa, $U_y[x]$ denotes the interference power shown as follows

$$U_y[x] = P_x^c I_{x,y}[x] + \sum_{y' \neq y} \rho_y[x] P_y^d[x] I_{y',y}[x] \tag{4}$$

where $I_{x,y}[x]$ is the channel between the y-th V2V receiver and the x-th V2I link over the x-th sub-band, $I_{y',y}[x]$ is the channel between the y-th V2V receiver and y'-th V2V transmitter over the x-th sub-band. Every V2V link is assumed to only access one sub-band, i.e., $\sum_x \rho_y[x] \leq 1$.

Capacities of the x-th V2I link and the y-th V2V link over the x-th sub-band are expressed as follows

$$C_x^c[x] = W \log(1 + S_x^c[x]) \tag{5}$$

$$C_y^d[x] = W \log(1 + S_y^d[x]) \tag{6}$$

where W denotes the bandwidth of each spectrum sub-band.

On the one hand, V2I links sum capacity needs to be maximized because the V2I links support high data rate mobile communication services. On the other hand, V2V links are designed to support sending and dependably receiving important messages. In [10], such a requirement is mathematically modeled as the transmit rate of size B packets within time T

$$\Pr\left\{ \sum_{t=1}^{T} \sum_{x=1}^{X} \rho_y[x] C_y^d[x,t] \geq \frac{B}{\Delta_T} \right\}, y \in \mathcal{Y} \tag{7}$$

where $C_y^d[x,t]$ denotes the capacity of the y-th V2V link, B is the V2V payload size, and Δ_T is the channel coherence time. The objective is to maximize the payload delivery rate of V2V links and the sum capacity of all V2I links.

Because of the mobility, a distributed V2V resource allocation scheme is better than a centralized controlling scheme in a vehicular environment. Then, it comes to a big challenge that how to coordinate the different actions of all V2V links instead of acting selfishly in their own interests. To figure out this problem, we propose deep reinforcement learning-based V2V spectrum allocation algorithms in the next section.

3. SAC-Based Resource Allocation

Unlike the traditional distributed optimization algorithm, reinforcement learning can solve the sequential decision-making problem in the dynamic vehicle networking environment, and enable the agent to explore and seek the effective strategy with the largest return using trial and error in the unknown vehicular environment. SAC algorithm is a reinforcement learning method proposed by T. Haarnoja et al. [14] in 2018 based on the idea of AC. Its main idea is to add entropy information based on the original reward to encourage exploration, and then train a behavior strategy with entropy to maximize the reward. Because it retains the randomness of behavior strategy to the greatest extent, it improves the agent's perception of the environment and enables the agent to adaptively adjust the strategy in the vehicle networking environment with changing channel conditions, which is more conducive to make reasonable spectrum selection. Due to the above advantages of SAC, this paper uses the idea of SAC to propose a new dynamic spectrum resource allocation algorithm for IoV [10,15].

To do this, it is necessary to establish a Markov decision process model (MDP) [16], which is shown in Figure 2, and in which each V2V link acts as an agent to interact with the environment in real time, that collects the state information of each time in the environment, makes decisions according to the vehicle conditions and requirements of the current environment, and obtains rewards.

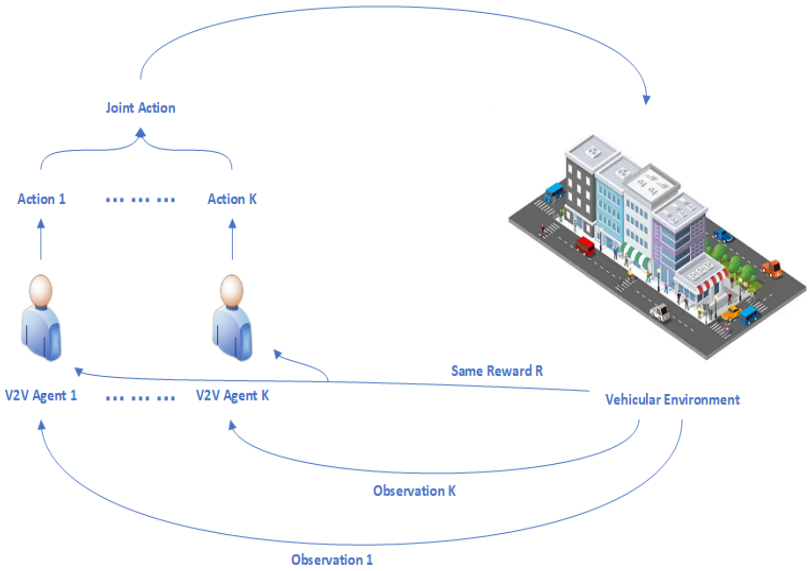


Figure 2. The interaction of multi-agent reinforcement learning (RL) formulation in vehicular environment.

3.1. State and Observation Space

As an agent, each V2V link y explores the unknown vehicular environment in the multi-agent resource allocation scenario [17,18] and the resource allocation problem can be mathematically seen as an MDP. In Figure 2, given the current state S_t, each V2V agent k obtains an observation function $Z_t^{(y)} = O(S_t, y)$ of the environment at each time step t, and then chooses an action $A_t^{(y)}$ to form a joint action A_t. Subsequently, each agent receives the same reward R_{t+1} and the vehicular environment enters to the next state S_{t+1} with probability $P(S_{t+1}, R_{t+1} \mid S_t, A_t)$. Then, each V2V agent will receive the next observation S_{t+1}.

Including all agents' behaviors and channel conditions, the real environment state S_t is unknown to each agent. They can only observe the environment through an observation function. The observation space of V2V agent Y includes $G_y[x]$, $I_{y',y}[x]$, $I_{y,B}[x]$, the interference channel from all the V2I transmitters $I_{x,y}[x]$, $I_y[x]$, the remaining payload of V2V B_x and the remaining time budgets T_x.

Apart from $I_{y,B}[x]$, such channel information can be accurately estimated by the y-th V2V link receiver at the start of time slot t [19]. Therefore, the observation function can be defined as following:

$$O(S_t, y) = \left\{ B_y, T_y, \{U_y[x]\}_{x \in X}, \{I_y[x]\}_{x \in X} \right\} \tag{8}$$

where $I_y[x] = \left\{ G_y[x], I_{y',y}[x], I_{y,B}[x], I_{x,y}[x] \right\}$.

3.2. Action Space

In the spectrum resource allocation of a multi-agent vehicle network, the V2V agent mainly selects the sub-band and its own transmit power. In this paper, the transmit power is chosen from −100 dBm to 23 dBm. The action function is defined as

$$A_t^{(k)} = (\text{transmit power } i, \text{sub-band } x). \tag{9}$$

3.3. Reward Design

The advantage of reinforcement learning for solving the optimization problem is the reward design. With the appropriate reward design, the system will be trained at each episode and its performance will be improved. As we mentioned, the objectives in this paper include maximizing the sum V2I capacity and improving the transmission success probability of V2V links within a constraint time T.

To achieve the first goal, we sum the capacities of all V2I links, $\sum_{x\in\mathcal{X}} C_x^c[x,t]$, and the capacity of x-th V2V link defined in (5) is a part of the reward. To achieve the other objective, for each agent y, the reward Y_y is set as the V2V transmission rate until all the payload is delivered. After that, the reward Y_y is set as a constant number φ, which is bigger than the largest V2V transmission rate. At each time step t, the V2V-related reward can be set as

$$\mathrm{Y}_y(t) = \begin{cases} \sum_{x=1}^{X} \rho_y[x] C_y^d[x,t], & \text{if } B_y \geq 0 \\ \varphi, & \text{otherwise.} \end{cases} \tag{10}$$

The elements of the reward function are not constants. Only when the agents finish the transmission, it will obtain a fixed reward. Under other different observations, the agents receive the different rewards. The agents do not receive the negative reward. When the agents perform a good action, it will receive a big reward, whereas when the agents take a not that good action, it will receive a small reward. The objective is to find an optimal allocation which can select an optimal action at each state that maximizes the total reward. It should be noted that the constant φ in the above reward function is set to balance the relationship between the final goal of training and the actual training efficiency. If only the final goal is considered, the agent will obtain zero reward before transmitting all the payload. However, in the actual training process, it is found that such a design will seriously hinder learning. The agents cannot learn anything useful because a sparse reward causes a problem that the agents keep obtaining zero reward in the early stage of training which will lead to a bad system performance. To avoid this case, we add some prior experience into the reward. In practice, φ is a hyper parameter which needs to be set empirically. In the training process, φ is bigger than the largest transmission rate of V2V while φ should be less than twice the largest V2V transmission rate according to the training experience. Therefore, we design the second part of reward in (10) to solve this issue.

As a result, to achieve the two objectives, the reward function can be design as

$$R_{t+1} = \varphi_c \sum_x C_x^c[x,t] + \varphi_d \sum_y \mathrm{Y}_y(t) \tag{11}$$

where φ_c and φ_d are positive weights to balance V2V and V2I goals. It should be noted that all V2V receive the same reward so that cooperative policy among all the agents is encouraged.

4. Learning Algorithm

In this paper, SAC is used to solve the spectrum resource allocation problem. The basic idea is described as follows: firstly, the policy network generated by the approximate action strategy and the soft Q network judged by the policy value are established. Then, at each step, the state, next state, action, and reward generated by each V2V link in the IoV are stored as an experience in the memory. Finally, the network is trained in reverse by optimizing the network loss function to obtain a better resource allocation strategy [14,20].

The optimization objective of reinforcement learning is to find an optimal policy to achieve the cumulative return maximization. The optimization objective of the SAC algorithm is not only to maximize the cumulative reward, but also to maximize the entropy, which is shown as:

$$\pi^* = \arg\max_{\pi} \mathrm{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t(R(s_t,a_t,s_{t+1}) + \alpha\Gamma((\cdot \mid s_t)))\right] \tag{12}$$

with

$$\Gamma(P) = \mathrm{E}_{\epsilon\sim P}[-\log P(\epsilon)] \tag{13}$$

where Γ is the entropy denoting the degree of randomization of the current policy π, α is the temperature parameter balancing the weight of the reward and the stochasticity of the optimal policy, and γ is the discount parameter. The probability density function of random variable ϵ is P, so that the entropy of ϵ can be determined.

According to Bellman's general recurrence equation, the function of judging the value of the behavior strategy in reinforcement learning is

$$\mathrm{Q}^{\pi}(s_t,a_t) = \mathrm{r}(s_t,a_t) + \gamma \mathrm{E}_{s_{t+1},a_{t+1}}[\mathrm{Q}^{\pi}(s_{t+1},a_{t+1})]. \tag{14}$$

The Q value of state S_t taking action a_t is determined by the sum of reward r_t and the discount expectation of the Q value of state S_{t+1} taking action a_{t+1}. In SAC, the entropy of the policy shown in (13) also needs to be considered. Thereafter, the new Q value function is shown as:

$$\begin{aligned}\mathrm{Q}(s_t,a_t) =& \mathrm{r}(s_t,a_t) + \gamma \mathrm{E}_{s_{t+1},a+1}[\mathrm{Q}(s_{t+1},a_{t+1}) \\ & -\alpha\log_2(\pi(a_{t+1} \mid s_{t+1}))]\end{aligned} \tag{15}$$

where $\mathrm{E}_{s_{t+1},a_{t+1}}$ denotes the expectation value of the next state s_{t+1} from memory and the action a_{t+1} based on the current policy.

4.1. Policy Network

First of all, a policy neural network needs to be constructed as the actor network to generate action strategy. The input of the policy network is all the information of observation in the environment and the outputs are the probability of every action and the chosen action. The policy function is shown as:

$$A_t = \pi_{\phi}(S_t). \tag{16}$$

The loss function of the policy network can be defined as following [14]:

$$\mathrm{J}_{\pi}(\phi) = \mathrm{E}_{\mathrm{S}_t\sim\mathrm{S}}\left[\alpha\sum_{z=1}^{Z}\left[\pi(s_i)^{\mathrm{T}}\log_2\pi(s_t)\right] - \mathrm{Q}_{\theta}^{*}(s_i)\right] \tag{17}$$

with

$$\mathrm{Q}_{\theta}^{*} = \sum_{z=1}^{Z}\left[\pi(s_t)^{\mathrm{T}}\min\left(\mathrm{Q}_1(s_t),\mathrm{Q}_2(s_t)\right)\right]. \tag{18}$$

Two different critic target networks are used to avoid the overvalued problem of the estimation of the Q value. Therefore, there are two Q values, i.e., $\mathrm{Q}_1(s_t)$ and $\mathrm{Q}_2(s_t)$ in (18) where the smaller one will be chosen.

4.2. Soft Q Network

Secondly, in order to judge the policy network, we construct two soft Q neural networks, namely the current value network and target network. The inputs of both networks

are the state information of observation and the outputs of both networks are the Q value of the action shown as:

$$\mathrm{Q}(s_t, a_t) = \mathrm{r}(s_t, a_t) + \gamma \mathrm{E}_{z_{t+1} \sim \rho}\left[\mathrm{V}\left(s_{t+1}\right)\right]. \tag{19}$$

The soft value can be defined as:

$$\begin{aligned} \mathrm{V}_{\mathrm{soft}}\left(s_t\right) =& \pi(s_t)^{\mathrm{T}}[\min(\mathrm{Q}_1(s_t), \mathrm{Q}_2(s_t)) \\ & -\alpha \log_2(\pi(s_t))] \end{aligned} \tag{20}$$

where $\pi(s_t)$ denotes the probability of every action based on state s_t. Then, the loss function of the soft Q current network can be expressed as:

$$\begin{aligned} \mathrm{J}_{\mathrm{Q}}(\theta) =& \frac{1}{x} \sum_{\mathrm{i}=1}^{x}[(\mathrm{Q}_\theta(s_t, a_t) - (\mathrm{r}(s_t, a_t) \\ & + \gamma \sum_{z=1}^{Z}[\,\mathrm{V}_\theta(s_{t+1})]))^2]. \end{aligned} \tag{21}$$

In order to further consider the impact of the temperature parameter α, we compare the fixed temperature parameter and the adaptive temperature parameter [20]. The latter one can automatically adjust depending on the environment and network learning. The loss function of the temperature parameter α can be defined as:

$$\mathrm{J}(\alpha) = -\mathrm{E}_{s_i \sim S}\left[\alpha\left(\sum_{z=1}^{Z}\left[\pi(s_t)^{\mathrm{T}} \log_2 \pi(s_t)\right] + \overline{\mathrm{H}}\right)\right] \tag{22}$$

where $\overline{\mathrm{H}} = -\dim(as)$ is the dimension of the action space defined as the practical experience [14].

4.3. Memory Buffer

SAC algorithm is similar to the Q-learning method in reinforcement learning, and it adopts the idea of experience replay. In the training process, using all sample data to participate in training will cause a slow training speed and poor convergence, whereas using part of sample data cannot achieve a good training effect. The above problem can be solved by establishing two different fixed-size buffer memories. All the experience is stored in the big memory buffer and the high temporal difference (TD) error experience is stored in the small memory buffer. The experience in the small one will be chosen to train more times for its high TD error which can accelerate the training process. When the memory is full, the memory is released to reserve space for new sample data. By setting the capacity M of the memory bank to control the data level involved in the training, most of the sample data are retained as much as possible to achieve good training results. In the actual training process, to speed up the training speed, the experience is randomly selected from the memory in batches for the reverse training of the network.

4.4. Network Learning

The essence of network learning is the process of constantly updating network parameters to minimize the network loss function. In Algorithm 1, the gradient update and soft update are used to update the network. Specifically, the gradient update method is used to update the parameters and temperature entropy coefficient of the soft Q current network and policy network, and the updating functions are shown as:

$$\begin{aligned} \theta^{Q} &= \theta^{Q} + \lambda \nabla_{\mathrm{J}}(\theta) \\ \phi^{\pi} &= \phi^{\pi} + \lambda \nabla_{\mathrm{J}}(\phi) \\ \alpha &= \alpha + \lambda \nabla_{\mathrm{J}}(\alpha) \end{aligned} \tag{23}$$

where λ is the gradient update weight, θ^{Q} and ϕ^{π} are the network parameters of the soft Q current network and policy network, respectively.

Algorithm 1 Resource sharing based on SAC in multi-Agent reinforcement learning.

Generate vehicular environment Z_t and initialize all parameters
for all V2V agent y **do**
    Initialize actor, critic, critic target for all agents randomly
    Initialize memory buffer F_t
end for
each episode
Update all vehicles location, large-scale fading
Reset B_x = B and T_x = T, for all $y \in \mathcal{Y}$
for each step t **do**
    for each V2V agent y **do**
        Observe the vehicular environment Z_t
        Choose action A_t from action space according to policy
    end for
    All agents take actions according to the policy
    Then receive reward R_t
    Update channel small-scale fading
    for each V2V agent y **do**
        Observe
        Store in the memory buffer F_t
    end for
end for
for each V2V agent y **do**
    Randomly sample some mini-batches from
    memory buffer F_t
    Update critic network and policy
    Update target network
end for

However, the soft Q target network does not participate in the learning progress, so it cannot update itself independently. Therefore, we choose soft update to copy the latest network parameters of the soft Q current network at regular intervals for small-scale updates, shown as follows:

$$\bar{Q} \leftarrow \eta Q + (1 - \eta)\bar{Q} \tag{24}$$

where η is the soft update weight and $\bar{Q}$ denotes the Q value of the target value network. The training process is shown in Algorithm 1.

However, when the vehicular environment becomes more sophisticated with more cars (the number of cars > 30), it is difficult for each agent with five networks to converge. Thus, we present Algorithm 2 to deal with a more complex case. In Algorithm 2, only one policy network (actor network) and one Q network (critic network) are needed to be trained with the parameter sharing in this learning problem [21]. At each episode, only one agent interacts with the vehicular environment. Networks will be trained by the agent experience and then the next agent will succeed the networks and continue to train. Algorithm 2 greatly reduces the complexity of the networks and it is easier to converge.

Algorithm 2 Sharing soft actor critic for resource allocation in V2X.

Generate vehicular environment and initialize all parameters
for all V2V agent y **do**
    Initialize actor, critic, critic target for all agents randomly
    Initialize memory buffer F_t
    Initialize temperature Z_t
end for
for each V2V agent y **do**
    for each episode **do**
        Update all vehicles location, large-scale fading
        Reset B_x = B and T_x = T, for all $y \in \mathcal{Y}$
        for each step **do**
            Observe
            Choose action A_t from action space according to policy
            Receive reward R_t
            Update channel small-scale fading
            Observe
            Store in the replay memory buffer F_t
            Randomly sample some mini-batches from
            memory buffer
            Update critic network
            Update policy network
            Update temperature Z_t
            Update target network
            Update the observation
        end for
    end for
end for

5. Simulation Results

The simulation considers the vehicle networking topology scenario in the area of 375 m wide and 649 m long one-way lanes. A base station is set on the left of the scenario. Other simulation parameters of the system can be referred to as 3GPP tr 36.885 [21,22], which describes the vehicle drop models, densities, speeds, direction of movement, vehicular channels, V2V data traffic, etc., shown in Table 1. The rectified linear unit is used as the activation function and the RMSProp optimizer is used to update the network parameter. Each V2V agent consists of five fully connected hidden neural networks in which the number of neurons in the three hidden layers is set to 500, 250, and 125, respectively, and we train each agent's networks with 4000 episodes. In both algorithms, we set the gradient update rate of the soft Q current network and the policy network to 0.001 and the soft update rate of soft Q target network to 0.01. In Algorithm 1, the temperature parameter α is fixed for which we set it to 0.001, while in Algorithm 2, the temperature parameter is adaptive and we set it to 0.95. Thereafter, we compare different SAC algorithms with DQN in the simulation to show the better performance.

Firstly, in order to verify the effectiveness of the SAC algorithm, the convergence of the algorithm must be verified. As we can see in Figures 3–5, with the increase in training times, the cumulative reward sum returned by each training set gradually increases and finally tends to converge, which verifies the effectiveness of the two SAC algorithms. Additionally, the soft actor critic-fix (SAC-fix) converges to about 110 return; the PSSAC converges to about 118 return; and soft actor critic learn (SAC-learn) converges to about 120 return. This means that SAC-learn and PSSAC perform better than SAC-fix.

Secondly, the V2V payload transmission success probability and V2I sum capacity of the PSSAC, SAC-fix, SAC-learn, and DQN are compared. When the number of vehicles is 6, the V2I sum capacity and V2V payload transmission success probability of the two SAC algorithms and DQN algorithm are simulated and compared in Figure 6a,b, respectively.

This shows that, compared with the SAC algorithm, the V2I sum capacity of the DQN algorithm is better when the size of the loads is small, but the performance obviously decreases with the increase in the size of loads. In addition, the V2I performance of the SAC algorithm is inferior to that of the DQN algorithm, but when the required transmission load increases to 8 × 1060 bytes, the performance of the SAC algorithm begins to exceed that of DQN algorithm, and the performance changes more stably with the increase in loads. In Figure 7a,b, when the vehicle number comes to 10, the advantages of the DQN algorithm are no longer prominent and SAC algorithms always perform better than DQN with the increase in payload size. In addition, when the vehicular environment becomes more complex, such as increasing the payload size and vehicle size, the SAC-learn performs better than others.

Table 1. Simulation parameters.

Symbol	Quantity	Value
X	Number of V2I links	4 6 8 10 20
Y	Number of V2V links	4 6 8 10 20
C	Carrier frequency	2 GHz
W	Bandwidth	4 MGz
BG	BS antenna gain	8 dBi
BH	BS antenna height	25 m
BN	BS receiver noise	5 dB
VH	Vehicle antenna height	1.5 m
VG	Vehicle antenna gain	3 dBi
VN	Vehicle receiver noise	9 dB
v	Absolute vehicle speed	36 km/h
σ^2	Noise power	−114 dBm
P^d	V2V transmit power	−100~23 dBm
P^c	V2I transmit power	23 dBm
T	Time constraint of V2V payload transmission	100 ms
B	V2V payload size	[1, 2, ...] × 1060 bytes
	Vehicle drop and mobility model	Urban case of A.1.2 in [22]

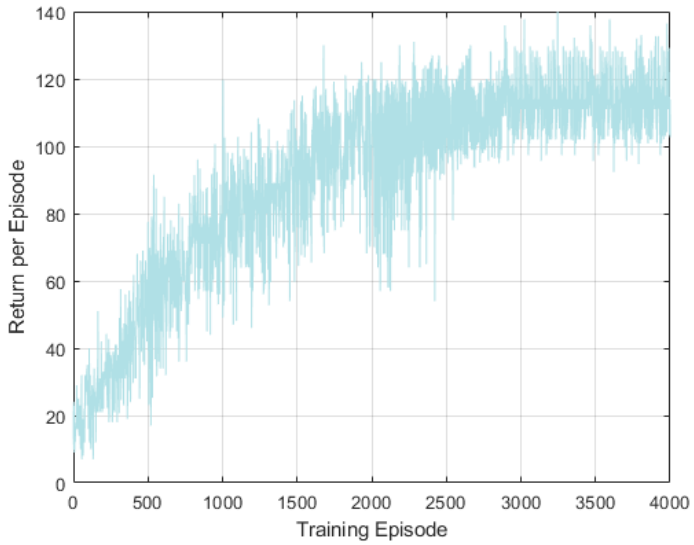


Figure 3. Convergence performance of SAC-fix.

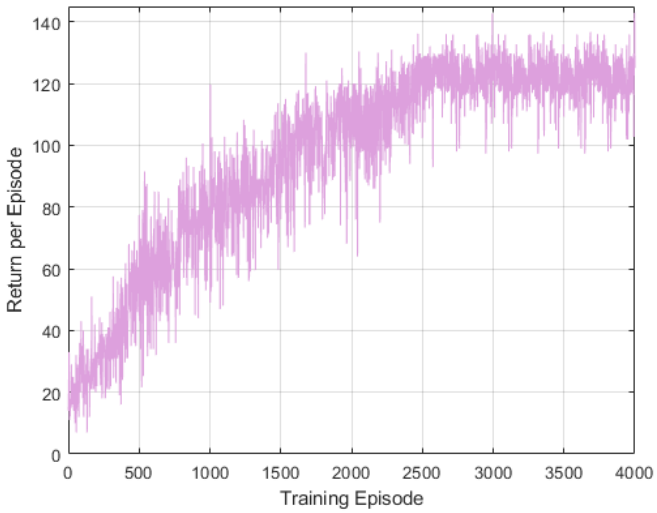


Figure 4. Convergence performance of SAC-learn.

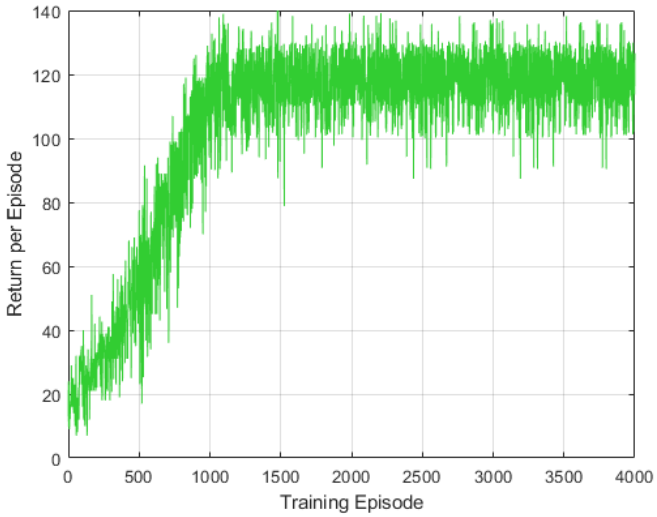
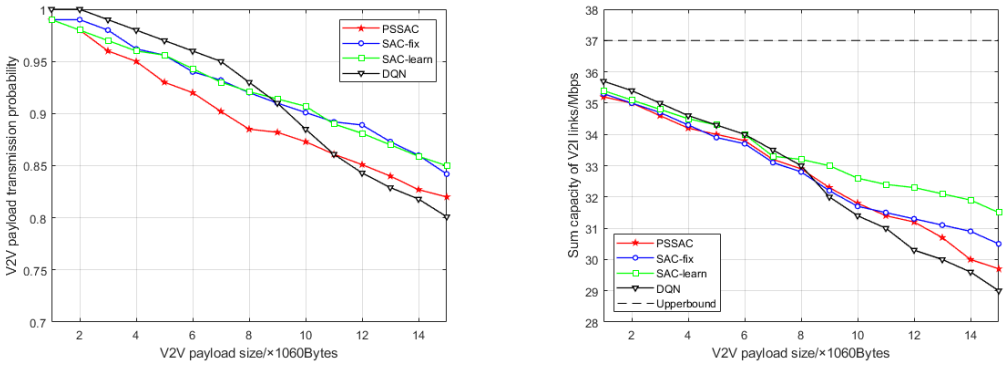


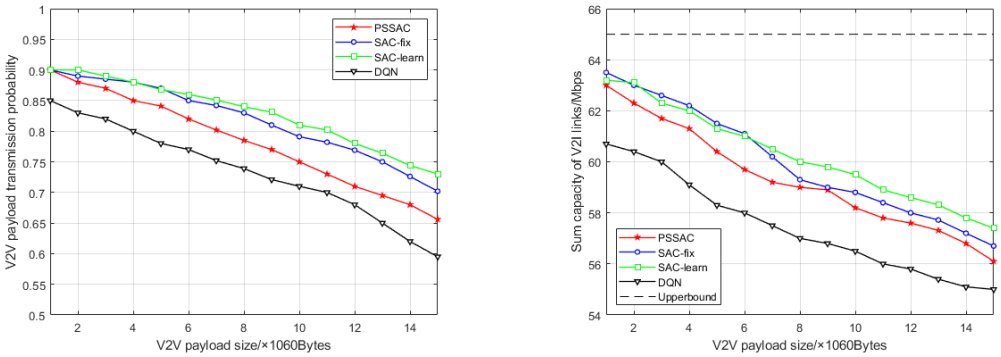
Figure 5. Convergence performance of PSSAC.



(**a**) (**b**)

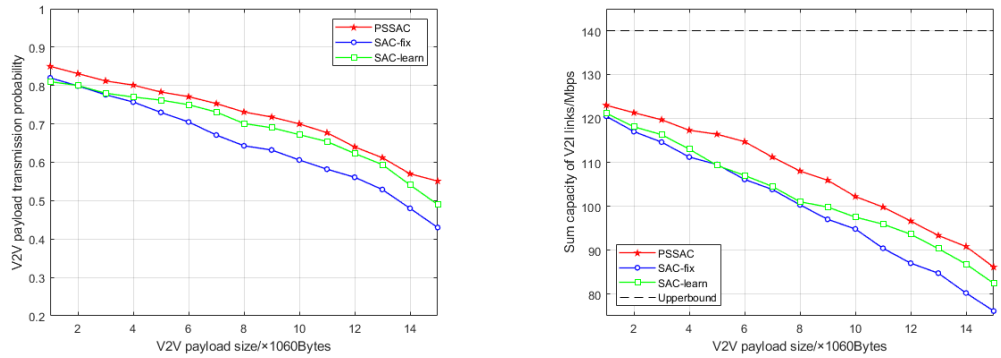
Figure 6. Vehicle number = 6: (**a**) V2V payload transmission success probability with the varying payload; and (**b**) sum capacity of V2I links with the varying V2V payload size B.

In Figure 8a,b, it is obvious that, when the vehicle number comes to 20, PSSAC always performs best. Because of the complexities of the SAC-fix and SAC-learn algorithms, it is difficult to converge to a good return and sometimes barely converge. Because DQN cannot deal with such a complex situation, it is not included in the simulations. If the number of V2V connections increases, the performance of all algorithms will get worse and converge more slowly. In addition, in real-world scenarios, there are more details we should consider, such as weather. Therefore, the reward function and environment need redefining.



(**a**) (**b**)

Figure 7. Vehicle number = 10: (**a**) V2V payload transmission success probability with the varying payload; and (**b**) sum capacity of V2I links with the varying V2V payload size B.



(**a**) (**b**)

Figure 8. Vehicle number = 20: (**a**) V2V payload transmission success probability with the varying payload; and (**b**) sum capacity of V2I links with the varying V2V payload size B.

In addition, in order to see how different agents transmit and cooperate in the transmit payload process, we record the remaining payload of each agent. When the number of vehicles is 6, the residual load changes of the V2V link in the SAC-fix algorithm and DQN algorithm are also observed, as shown in Figure 9a,b, respectively. On the one hand, both algorithms complete the load transmission task in a very short time. On the other hand, the DQN algorithm takes about twice the amount of time as the SAC-fix algorithm does and its stability is worse than the SAC-fix algorithm. Because the adoption of the maximum entropy idea of the SAC algorithm makes the agent randomize the action as much as possible—on the basis of completing the current task—to obtain a variety of approximate optimal choices, this improves the agent's exploration ability in the environment and improves the stability of SAC algorithm in the dynamic environment.

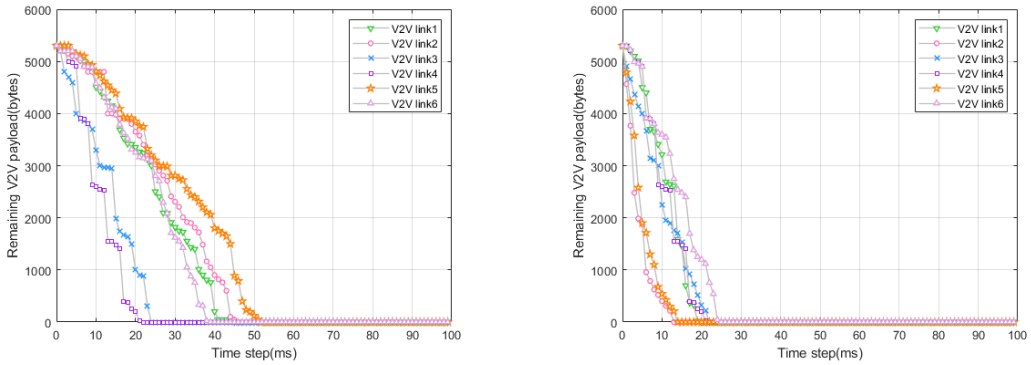


Figure 9. (**a**) Remaining payload of each agent (DQN); and (**b**) remaining payload of each agent (SAC-fix).

6. Conclusions

In this paper, we solved the existing problems of the spectrum allocation in IoV. Firstly, we proposed an SAC-based scheme to solve the spectrum allocation problem in the complex vehicular environment which has a better performance than the DQN scheme. In addition, we proposed the other SAC-based scheme to reduce the complexity to easy convergence, which reduces the training time by about 50 percent. These novel multi-agent SAC schemes (SAC-fix, SAC-learn, PSSAC) have 10 percent performance improvements in terms of the V2I sum capacity and the V2V payload transmission success probability compared with DQN in the vehicular environment. In the real ITS environment, some elements have not been considered in our scenario and algorithm, such as the weather, passersby, and vehicle density. In the future, we will consider more details of the vehicle network and make the V2V environment more realistic, and further improve our algorithm.

Author Contributions: Conceptualization, H.W. and Y.P.; methodology, H.W.; validation, H.W.; formal analysis, H.W.; investigation, H.W.; writing—original draft preparation, H.W. and Y.P.; writing—review and editing, M.Y. and Y.P.; visualization, H.W.; supervision, Y.P.; project administration, Y.P.; funding acquisition, Y.P., F.A.-H., M.M.M. and J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by The Science and Technology Development Fund, Macau SAR (0108/2020/A3), (0005/2021/ITP), (0066/2020/A2), (0047/2020/A1), and (0014/2022/A); and in part by Wuyi University-Hong Kong-Macau joint Research and Development Fund (2021WGALH17).

Data Availability Statement: Data sharing not applicable.

Acknowledgments: The researchers would like to acknowledge the Deanship of Scientific Research at Taif University for funding this work.

Conflicts of Interest: The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

1. Liao, X.; Yan, S.; Shi, J.; Tan, Z.; Zhao, Z.; Li, Z. Deep reinforcement learning based resource allocation algorithm in cellular networks. *J. Commun.* **2019**, *40*, 11–18.
2. Min, Z.; De-min, L.; Kang, J. Research of dynamic channel allocation algorithm for multi-radio multi-channel VANET. *Appl. Res. Comput.* **2014**, *31*, 1516–1519.
3. Xue, L.; Fan, X. Cognitive Spectrum Allocation Mechanism in Internet of Vehicles Based on Clustering Structure. *Comput. Sci.* **2019**, *9*, 143–149.
4. Sun, W.; Ström, E.G.; Brännström, F.; Sou, K.C.; Sui, Y. Radio resource management for D2D-based V2V communication. *IEEE Trans. Veh. Technol.* **2016**, *65*, 36–50. [CrossRef]
5. Sun, W.; Yuan, D.; Ström, E.G.; Brännström, F. Cluster-based radio resource management for D2D-supported safety-critical V2X communications. *IEEE Trans. Wirel. Commun.* **2016**, *15*, 2756–2769. [CrossRef]

6. Wang, Z.; Zhang, Y.; Yin, C.; Huang, Z. Multi-agent Deep Reinforcement Learning based on Maximum Entropy. In Proceedings of the 2021 IEEE 4th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), Chongqing, China, 18–20 July 2021; pp. 1402–1406.
7. Salahuddin, M.A.; Al-Fuqaha, A.; Guizani, M. Reinforcement learning for resource provisioning in the vehicular cloud. *IEEE Wirel. Commun.* **2016**, *23*, 128–135. [CrossRef]
8. He, Y.; Zhao, N.; Yin, H. Integrated networking, caching and computing for connected vehicles: A deep reinforcement learning approach. *IEEE Trans. Vel. Technol.* **2018**, *67*, 44–55. [CrossRef]
9. Chen, M.; Chen, J.; Chen, X.; Zhang, S.; Xu, S. A deep learning based resource allocation scheme in vehicular communication systems. In Proceedings of the 2019 IEEE Wireless Communications and Networking Conference (WCNC), Marrakesh, Morocco, 15–18 April 2019; pp. 1–6.
10. Liang, L.; Ye, H.; Li, G.Y. Spectrum Sharing in Vehicular Networks Based on Multi-Agent Reinforcement Learning. *IEEE J. Sel. Areas Commun.* **2019**, *37*, 2282–2292. [CrossRef]
11. Zhang, H.; Wang, Z.; Liu, K. V2X offloading and resource allocation in SDN-assisted MEC-based vehicular networks. *China Commun.* **2020**, *17*, 266–283. [CrossRef]
12. Xu, Y.H.; Yang, C.C.; Hua, M.; Zhou, W. Deep Deterministic Policy Gradient (DDPG)-based resource allocation scheme for NOMA vehicular communications. *IEEE Access* **2020**, *8*, 797–807. [CrossRef]
13. Haarnoja, T.; Zhou, A.; Hartikainen, K.; Tucker, G.; Ha, S.; Tan, J.; Kumar, V.; Zhu, H.; Gupta, A.; Abbeel, P.; et al. Soft actor-critic algorithms and applications. *arXiv* **2018**, arXiv:1812.05905.
14. Haarnoja, T.; Zhou, A.; Abbeel, P.; Levine, S. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. *arXiv* **2018**, arXiv:1801.0129v2.
15. Ye, H.; Li, G.; Juangi, B. Deep reinforcement learning based resource allocation for V2V communications. *IEEE Trans. Veh. Technol.* **2019**, *68*, 3163–3173. [CrossRef]
16. Buşoniu, L.; Babuška, R.; Schutter, B.D. *Multi-Agent Reinforcement Learning: An Overview*; Springer: Berlin/Heidelberg, Germany, 2010.
17. Foerster, J.; Nardelli, N.; Farquhar, G.; Afouras, T.; Torr, P.H.; Kohli, P.; Whiteson, S. Stabilising experience replay for deep multi-agent reinforcement learning. In Proceedings of the International Conference on Machine Learning (ICML), Volterra, Tuscany, Italy, 6–11 August 2017; pp. 1146–1155.
18. Omidshafiei, S.; Pazis, J.; Amato, C.; How, J.P.; Vian, J. Deep decentralized multi-task multi-agent reinforcement learning under partial observability. In Proceedings of the International Conference on Machine Learning (ICML), Volterra, Tuscany, Italy, 6–11 August 2017; pp. 2681–2690.
19. Gupta, J.K.; Egorov, M.; Kochenderfer, M. Cooperative multi-agent control using deep reinforcement learning. In Proceedings of the International Conference on Autonomous Agents and Multiagent Systems, Atlanta, GA, USA, 8–12 May 2017; Springer: Cham, Switzerland; pp. 66–83.
20. Oddi, G.; Panfili, M.; Pietrabissa, A.; Zuccaro, L.; Suraci, V. A resource allocation algorithm of Multi-cloud resources based on Markov decision process. In Proceedings of the IEEE 5th International Conference on Cloud Computing Technology and Science, Bristol, UK, 2–5 December 2013; pp. 130–135.
21. WF on SLS Evaluation Assumptions for EV2X. Document R1-165704, 3GPP TSG RAN WG1 Meeting. May 2016. Available online: https://www.3gpp.org/ftp/tsg_ran/WG1_RL1/TSGR1_85/Docs/R1-165704.zip (accessed on 27 July 2023).
22. 3rd Generation Partnership Project. Technical Specification Group Radio Access Network. Study on LTE-Based V2X Services (Release14); 3GPP TR 36.885 V14.0.0. 2016. Available online: https://portal.3gpp.org/Meetings.aspx#/meeting?MtgId=31638 (accessed on 27 July 2023).