IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 17, 2019, accepted December 27, 2019, date of publication December 30, 2019, date of current version January 7, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2963051

Contract-Based Computing Resource Management via Deep Reinforcement Learning in Vehicular Fog Computing

JUNHUI ZHAO[1,2], **(Senior Member, IEEE), MING KONG**[2], **QIUPING LI**[2], **AND XIAOKE SUN**[2]

[1]School of Information Engineering, East China Jiaotong University, Nanchang 330013, China
[2]School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China

Corresponding author: Junhui Zhao (junhuizhao@hotmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61661021 and Grant 61971191, in part by the Beijing Natural Science Foundation under Grant L182018, in part by the National Science and Technology Major Project of the Ministry of Science and Technology of China under Grant 2016ZX03001014-006, in part by the Open Research Fund of National Mobile Communications Research Laboratory, Southeast University, under Grant 2017D14, and in part by the Jiangxi Provincial Cultivation Program for Academic and Technical Leaders of Major Subjects under Grant 20172BCB22016.

ABSTRACT Vehicle fog computing (VFC) is proposed as a solution that can significantly reduce the task processing overload of base station during the peak time, where the vehicle as a fog node contributes idle computing resource for task processing. However, there are still many challenges in the deployment of VFC, such as the lack of specific incentives of resource contribution, high system complexity, and offloading collisions between vehicles when the vehicles are offloading tasks simultaneously. In this paper, we first propose a novel contract-based incentive mechanism that combines resource contribution and resource utilization. Based on that, we propose to use distributed deep reinforcement learning to allocate resources and reduce system complexity. Task offloading method based on the queuing model is also proposed to avoid decision collisions in multi-vehicles task offloading. Numerical experiment results demonstrate that our proposed scheme has achieved a significant improvement in task offloading and resource allocation performance.

INDEX TERMS Vehicular fog computing, contract theory, deep reinforcement learning, resource allocation, task offloading.

I. INTRODUCTION

With the rapid development of the Internet of Vehicles(IoV) technology and 5G communication technology, more and more functional technologies are applied to vehicles, such as augmented reality(AR), real-time video streaming, automatic driving(AD), etc. [1]–[3]. In these applications, some of them need to transmit a large amount of data, others do not need to transmit a lot of data, however, there would be rigorous delay constraints to transmit them, so these applications have a relatively large demand for the IoV resources, such as spectrum resources, storage space, etc. But local vehicle capability is limited and vehicle would be too hard to accomplish its task. Therefore, cloud computing with strong computing capability has become a more efficient way to implement these applications [4]. The data that needs to be processed by these applications is transmitted to the cloud server for processing, and the cloud server sends the processing result to the vehicle end to complete the task processing [5].

However, long-distance data transmission may create some potential challenges, such as large data transmission delay cannot guarantee the quality of service (QoS). Extending the computation resources of the cloud computing to the mobile edge computing (MEC) close to mobile users is proposed [6]. The MEC puts the cloud services to the radio access network (RAN) and offers the cloud-computing capability in close proximity to mobile users [7].

The research on MEC mainly focuses on how to optimize the decision of task offloading and the strategy of resource allocation to improve the performance of the IoV. In [8], the author's optimization goal is to minimize the overall energy consumption of the system and the use of computation

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

resources. The work in [9]–[15] is to minimize the delay for all users and the energy consumption of them by proposing a novel IoV systems and resource allocation mechanisms.

Although MEC has strong computing power, with the increase of the number of vehicles, its limited computing power is gradually overloaded, which makes the QoS of some vehicles cannot be guaranteed. On the other hand, some vehicles are equipped with processors with strong computing power, but the processing unit of the vehicle is idle [16], [17]. Therefore, the use of idle computation resources of these vehicles can significantly alleviate the load pressure of the MEC without additional deployment of the MEC. In [18], the parked vehicles can be used as a fog node and provide real-time computation resources. At the same time, the vehicles can directly offload tasks to vehicles with idle computation resources for task processing, improving task processing efficiency. This method is called vehicle fog computing (VFC) [19].

Neverthless, VFC still faces some challenges. In [20], vehicles with idle computation resources unconditionally contribute computation resources to the IoV. However, in actual scenarios, vehicles are privately owned and require certain incentives. The incentive vehicles contribute computation resources, that is, the VFC gives incentives to vehicles that contribute resources. These vehicles get these rewards so that they can use the system resources (storage space, frequency resource, etc.) of the IoV. The more resources the vehicles contribute, the more rewards the vehicles would obtain. In [21]–[23], an optimization algorithm for computation resource allocation and task unloading is proposed, where the system gives certain rewards to vehicles that contribute resources, but the article does not clearly indicate the purpose of the reward. Therefore, it is necessary to design a reasonable incentive mechanism to encourage vehicles to contribute resources. At the meantime, vehicles can use these rewards to exchange vehicle network system resources to improve the system performance.

The employment of conventional computation offloading and resource allocation in MEC and VFC makes the system complex in [3], where the offloading decisions are taken through game-theoretic approach and the resource allocation is achieved by using the Lagrange multiplier method. In order to reduce the system implementation complexity, more and more researches have focused on the scenarios that use deep reinforcement learning (DRL) algorithms, in which deep neural network (DNN) is introduced, which will be trained to make offloading decisions and resource allocation instead of using other methods, to optimize offloading decisions and resource allocation to achieve system optimization goals in recent years.

In [24], the method of DRL is used to meet different resource requirements. A method based on DRL is studied in [25], which simplifies the state of the system for distributed offloading. The author in [26] proposes a DRL based transmission strategy by exploring trirelationships among vehicles. In [27], the author constructs an intelligent offloading system for vehicular edge computing by leveraging DRL. In [28], the DRL is used to save energy in RAN while meeting the needs of users. The author in [29] proposes a new DRL algorithm for solving the higher complexity joint resource management problem in the IoV. In [30], reinforcement learning is used to solve the problem of resource allocation in the vehicle cloud, in which resources can be dynamically allocated to maximize long-term network rewards and prevent myopic decisions.

In this paper, we propose a new contract theory that incents vehicles contribute their computation resources and get the rewards so that they can use them to exchange additional resources from the IoV to improve the QoS of their application, such as task processing delay, energy consumption, etc. We also use DRL method to reduce the system implementation complexity, which based on our proposed contract theory offloads the tasks of vehicles and reasonably allocates system resources to achieve better system performance. Our main contributions of this work are summarized as follows:

- We propose an incentive mechanism based on contract theory. While the vehicles contribute to the computation resources, the reward of the previous acquisition can be used to exchange the resources of the system, such as frequency resources, computation resources, etc., to improve the QoS of the vehicles application. When providing rewards and resources for the vehicles, the roadside unit (RSU) can obtain the idle computation resources of the vehicles to improve its own computing power and the performance of the entire system when the vehicles offload the task.
- We will use the DRL method based on incentive mechanism to reduce implementation complexity in VFC, where offloading decisions could be generated faster through DNN. Besides, in this new incentive mechanism, since there are some vehicles using additional resources, DRL can be used as an efficient method to allocate these resources in a low complexity way.
- The DRL method we adopt is a distributed algorithm. In order to avoid the task offloading conflict caused by the simultaneous offloading decision, we introduce a queuing model, which is sorted according to the accumulated rewards of the vehicles. The vehicle that has higher cumulative reward can obtain the priority of task offloading, avoiding the task offloading conflict caused by simultaneous decision.

The remaining parts of the paper are summarized as follows. The system model is introduced in Section II. In Section III, a new mechanism based contract theory is presented. We describe a framework of DRL based task offload and resource allocation in Section IV. In Section V, the simulation presented. Conclusions and future work are drawn in Section VI and Section VII, respectively.

II. SYSTEM MODEL

In this paper, the IoV scenario is shown in Fig. 1. This scenario consists of a RSU with powerful computing capacity and M vehicles denoted by $\mathcal{M} \in \{1, 2, 3 \ldots M\}$ moving on the road. The location of the RSU is p_0, and the computing capacity of the RSU is represented as C_0. Vehicles are evenly distributed on a dual carriageway with two lanes. The position and velocity of the vehicle is denoted as p_i and v_i, respectively, and the idle computation resources owned by vehicle is denoted as C_i. In the k^{th} time slot, the vehicle randomly generate task $\left(T_i^k, d_i^k\right)$ that need to be processed, where T_i^k represents the size of the task generated by the vehicle in the k^{th} time slot, and d_i^k represents the maximum tolerable delay of the task.

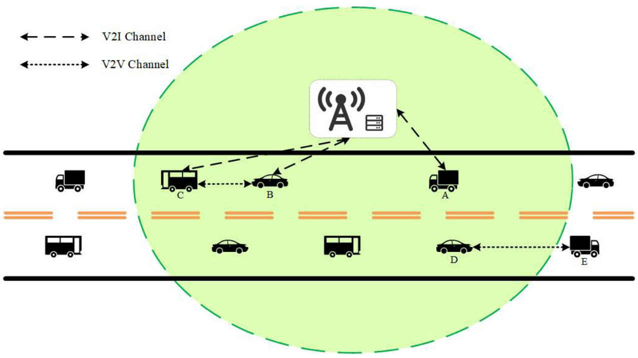

FIGURE 1. The scenario of the internet of vehicles.

In this scenario, the vehicle can directly perform vehicle-to-infrastructure (V2I) communication with the RSU, such as vehicle A, and the vehicle can also perform vehicle-to-vehicle (V2V) communication with other vehicles with the assistance of RSU, such as vehicles B and C. Even if the vehicle leaves the communication range of RSU but is within the communication range of other vehicles, V2V communication can still be maintained between vehicles, such as vehicles D and E.

For each vehicle, they can handle the task by processing it locally or offload the task to other place to process. In local processing, the time t^c required by the vehicle to process the task in the k^{th} time slot can be expressed as

$$t_i^{kc} = \frac{T_i^k}{C_i}, \tag{1}$$

when $t_i^{kc} < d_i^k$, it indicates that the vehicle's own computing capacity can meet the delay requirement of the processing task, and the vehicle will process the task locally. Conversely, when $t_i^{kc} > d_i^k$, it indicates that the vehicle cannot complete the task, so the task needs to be offloaded.

As the task is offloaded, the vehicle i can offload the task to the vehicle j for processing and transmit task data through the V2V link. In order to improve the spectral efficiency, we assume that the allocated channels are orthogonal to each other and do not interfere with each other.

In the wireless communication of the k^{th} time slot, the signal noise ratio(SNR) of vehicle i is expressed as

$$\Gamma_{ij}^k = \frac{P_i^k h_{ij}}{N_0}, \tag{2}$$

where P_i^k represents the wireless transmission power of the vehicle i, h_{ij} represents the wireless channel gain between the vehicle i and the vehicle j, and N_0 represents the noise power [31], [32]. The gain of the wireless channel can be expressed as

$$h_{ij} = \mathrm{P}_i + G_i^t - L_{ij} + G_j^r, \tag{3}$$

where P_i denotes the transmission power of the vehicle i, and G_i^t, G_j^r denote the transmission antenna gain of the vehicle i and the reception antenna gain of the vehicle j, respectively. L_{ij} represents the transmission loss, which we assume is the path loss of free space, and L_{ij} is given by

$$L_{ij} = 32.4 + 20\lg F + 20\lg \sigma_{ij}, \tag{4}$$

where F is the transmission frequency and σ_{ij} is the distance between the vehicles i and j

$$\sigma_{ij} = \left|p_i - p_j\right|. \tag{5}$$

Then the wireless transmission rate of vehicle i to vehicle j denoted as

$$r_{ij}^k = W \log_2\left(1 + \Gamma_{ij}^k\right), \tag{6}$$

where W is the transmission bandwidth [33]. The time taken to transfer the task is shown as

$$t_{ij}^{kt} = \frac{T_i^k}{r_{ij}^k}. \tag{7}$$

The time taken by vehicle j to process the task is

$$t_j^{kc} = \frac{T_i^k}{C_j}. \tag{8}$$

We assume that the size of computation result data fed back from the vehicle j to the vehicle i is very small, so the transmission delay of the transmission processing result is negligible.

The total time taken by processing task by offloading the task to vehicle j is expressed as

$$t_{ij}^k = t_{ij}^{kt} + t_j^{kc}, \tag{9}$$

when $t_{ij}^k < d_i^k$, it indicates that the vehicle i offloads the task to the vehicle j and vehicle $j$can complete the task processing within d_i^k. Conversely, when $t_{ij}^k > d_{ij}$, it illustrates that the time taken by processing task exceeds the maximum tolerable delay of the task. If the vehicle i cannot complete the task processing by offloading the task to another vehicle, the task needs to be offloaded to the RSU for processing.

The RSU has powerful computing capacity, but in order to process more tasks offloaded to the RSU, we assume that the total time taken for the task to be offloaded to the RSU

($j = 0$) is equal to the maximum tolerable delay of the task, which can be represented as

$$t_{i0}^{k} = T_{i0}^{kt} + t_{0}^{kc} = d_{i}^{k}. \tag{10}$$

The amount of computation resources that the task generated by vehicle i requires to process form the RSU is shown as

$$D_{i}^{k} = \frac{T_{i}^{k}}{t_{0}^{kc}} = \frac{T_{i}^{k}}{d_{i}^{k} - t_{i0}^{kt}}. \tag{11}$$

Although the RSU has rich computation resources and strong computing capacity, it is also limited in computation resources. When the task of offloaded to the RSU exceeds the total computation resources of the RSU, some tasks processing would fail, which is denoted as

$$\sum_{i=0}^{N} \lambda_i \cdot D_{i}^{k} > C_0, \tag{12}$$

where $\lambda_i = 1$ indicates that the vehicle i offloads the task to the RSU for processing, otherwise, $\lambda_i = 0$.

We assume that in the k^{th} time slot, the channel conditions of the vehicle remain unchanged, which the data transmission rate does not change. In the $k^{th} + 1$ time slot, the position of the vehicle is expressed as

$$p_{i}^{k+1} = p_{i}^{k} + t \cdot v_i, \tag{13}$$

where t is the time interval of each time slot. At the same time, the vehicle will also generate new tasks (T_{i}^{k+1}, d_{i}^{k+1}) that needs to processing.

III. CONTRACT THEORY-BASED INCENTIVE MECHANISM

In this chapter, we propose a contract theory-based incentive mechanism for the current IoV scenario to encourage vehicles to contribute their own idle computation resources.

A. THE CONTRACT OF RESOURCE ALLOCATION

We assume that vehicles within the RSU communication range are classified into N types of vehicles according to their shareable resources, each type corresponding to a contract $(\xi_n, R(\xi_n))$, where ξ_n represents the type of resource contribution of vehicles, which is given by

$$\xi_1 < \xi_1 < \xi_1 \cdots < \xi_n, \quad n \in (1, 2, 3 \cdots N). \tag{14}$$

And $R(\xi_n)$ represents the bonus value corresponding to the resource contribution ξ_n. The more vehicles contribute the computation resources and the greater the probability that the resource will be utilized by other vehicles, the higher the reward value will be.

RSU will provide the corresponding contract according to the type of each vehicle. The vehicle can choose to accept or reject the contract, that is, decide whether to contribute its own computation resources. When $\xi_n = 0$, it means that the vehicle does not contribute resources, and the corresponding reward value $R(\xi_n) = 0$. Next we will establish the utility functions for the RSU and the vehicle and formulate our optimization goals.

Due to the asymmetry of the information, the RSU only knows that there are a total of N types of vehicles, and it is not clear the amount of resources that the vehicle can contribute. Therefore, for vehicle i, the probability of belonging to the n-type contract is ρ_n, and $\sum_{n=1}^{N} \rho_n = 1$.

1) UTILITY MODEL OF RSU

One purpose of vehicle contribution computation resources is to ease the burden of RSU computing tasks, while RSU is also responsible for resource scheduling. Therefore, we assume that the RSU utility function can be improved by reducing task processing latency, reducing the energy consumed by task processing, increasing the task completion ratio within the system, and reducing the utilization of system frequency resources, which is denoted as

$$U_{i}^{RSU}(\xi_n) = I_{\xi_n} - R^{C}(\xi_n), \tag{15}$$

where I indicates the RSU performance improvement, which means that the performance of the task offload to the vehicle processing is compared with the performance of task offload to the vehicle processing, and is given by

$$I_{\xi_n} = \omega_E \cdot \Delta E^{c} + \omega_t \cdot \Delta t^{c}, \tag{16}$$

where E and t represent the change of energy and task processing time when the task T is processed by the RSU and the vehicle as

$$\Delta E^{c} = (\varepsilon_{RSU} - \varepsilon_{Veh}) \cdot T, \tag{17}$$

$$\Delta t^{c} = t_0 - t_i, \tag{18}$$

where ω_{RSU} and ω_{Veh} represent the energy consumed by the task of processing the unit data amount on the RSU and the vehicle i, respectively, while t_0 and t_i represent the total time that the task is offloaded to the RSU and the vehicle i for processing, respectively.

For N types of M vehicles covered by the RSU communication range, the RSU utility function is as follows

$$U^{RSU}(\xi_n) = \sum_{i=1}^{M} \sum_{n=1}^{N} \rho_n U_{i}^{RSU}(\xi_n). \tag{19}$$

2) UTILITY MODEL OF VEHICLE

For the vehicle, if the computation resources are contributed and the resources are utilized by other vehicles, the reward corresponding to contract of the type n,$(\xi_n, R(\xi_n))$ can be obtained, the utility function of the vehicle i in the contract of n is as follows

$$U_{i}^{Veh}(\xi_n) = \theta_n R^{C}(\xi_n) - \xi_n, \tag{20}$$

where θ_n represents the contract weight of type n, and a higher type θ_n has a larger weight.

B. THE CONTRACT OF RESOURCE UTILIZATION

For a vehicle with a task offloading requirement, assuming that the accumulated reward value of the vehicle in the early task processing is τ, the vehicle can exchange the wireless transmission bandwidth w of the system with the accumulated reward value τ. So we propose a second set of contracts $\left(R^E\left(w_g\right), w_g\right)$, which has a total of G types, which means that the vehicle uses the reward value $R^E\left(w_g\right)$ in exchange for additional w_g bandwidth resources, which can reduce the time spent on wireless transmission. For vehicles with task offloading requirements, the probability of selecting a contract of type g is ρ_g, and $\sum_{g=1}^{G}\rho_g = 1$.

For RSU, the utility function in the contract for resource utilization is expressed as

$$U_i^{RSU}\left(w_g\right) = I_{w_g}^{RSU} - R^E\left(w_g\right), \tag{21}$$

$$I_{w_g}^{RSU} = f^{RSU} - \omega_w w_g, \tag{22}$$

where f is a penalty value, indicating that if the task processing fails to be completed, the penalty value of f is given as $f < 0$, and if the task is successfully completed, $f = 0$.

Assuming that there are X vehicles with task offloading requirements within the coverage of the RSU communication range, the RSU utility function is as follows

$$U^{RSU}\left(w_g\right) = \sum_{i=1}^{M}\sum_{g=1}^{G}\rho_g U_i^{RSU}\left(w_g\right). \tag{23}$$

For the vehicle i which offloads the task, the bonus value is exchanged for additional bandwidth resources, and the time taken for the processing task can be reduced. So the utility function is expressed as

$$U_i^{Veh}\left(w_g\right) = I_{w_g}^{Veh} - R^E\left(w_g\right), \tag{24}$$

where I indicates the improvement of the task processing performance of the vehicle after using the additional spectrum resources, as shown below

$$I_{w_g}^{Veh} = \omega_{t^{kt}}\Delta t^{kt} + f^{Veh}, \tag{25}$$

where Δt^{kt} represents the change value of the transmission delay after using the additional transmission bandwidth. f is a penalty value, indicating that if the task fails to complete the task processing, the penalty value of f is given as $f < 0$, and if the task is successfully completed, $f = 0$.

In summary, under the conditions of the two contracts, the overall utility equation of the whole system can be expressed as

$$U^S\left(\xi_n, w_g\right) = U^{RSU}(\xi_n) + U^{RSU}\left(w_g\right) + \sum_{i=1}^{M} U_i^{Veh}\left(\xi_n, w_g\right), \tag{26}$$

$$U_i^{Veh}\left(\xi_n, w_g\right) = \sum_{n=1}^{N}\rho_n U_i^{Veh}(\xi_n) + \phi \cdot \sum_{g=1}^{G}\rho_g U_i^{Veh}\left(w_g\right), \tag{27}$$

where $\phi = 1$ indicates that the vehicle i has a task offloading requirement, otherwise $\phi = 0$. So the corresponding optimization problem is as follows

$$\begin{aligned}
&\max_{\xi_n, w_g} U^S\left(\xi_n, w_g\right)\\
&s.t.\ C1: \theta_n R^C\left(\xi_n\right) - \xi_n > 0,\\
&\quad C2: \theta_n R^C\left(\xi_n\right) - \xi_n > \theta_n R^C\left(\xi_{n'}\right) - \xi_{n'},\\
&\quad C3: U_i^{RSU}\left(w_g\right) > 0,\\
&\quad C4: U_i^{RSU}\left(w_g\right) > U_i^{RSU}\left(w_{g'}\right),\\
&\quad \forall n, n' \in N, \quad n \neq n',\\
&\quad \forall g, g' \in G, \quad g \neq g',
\end{aligned} \tag{28}$$

where $C1$, $C3$ represent the IR constraint, and $C2$, $C4$ represent the IC constraint, of which means are shown as follows

- Individual rationality (IR) constraint: In $C1$, the type n vehicle will get a nonnegative utility value if it selects the contract item $(\xi_n, R(\xi_n))$, which is the same as that in $C3$.
- Incentive compatibility (IC) constraint: In $C2$, the type n vehicle will get the maximum utility value if and only if it select the contract $(\xi_n, R(\xi_n))$ designed for its type, which is the same as that in $C4$.

IV. DEEP REINFORCEMENT LEARNING FOR RESOURCE MANAGEMENT

In this section, we will introduce a task offloading and resource allocation mechanism based on DRL, and explain the key parts of reinforcement learning and the architecture of deep Q-networks (DQN) in detail.

A. REINFORCEMENT LEARNING

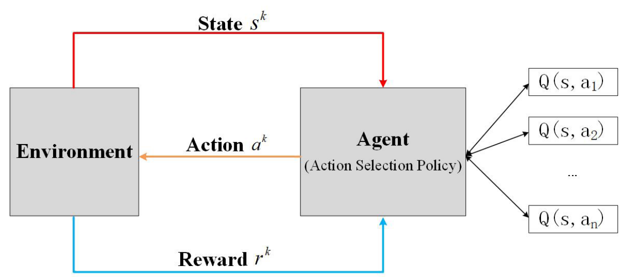

FIGURE 2. The architecture of reinforcement learning.

Fig. 2 shows the architecture of reinforcement learning, which contains an agent and environment, which interact with each other. In the IoV scenario, each vehicle is an agent, and the environment includes all vehicles and RSUs in the IoV scenario. In reinforcement learning, the agent maintains a Q value table $Q\left(s_i, a_j\right)$, where s represents the feature vector of the current environment, a represents the action taken by the agent, and Q value represents the action that can be obtained by taking action a in state s value. The agent selects action a^* according to the Q value table and transmits it to the environment.

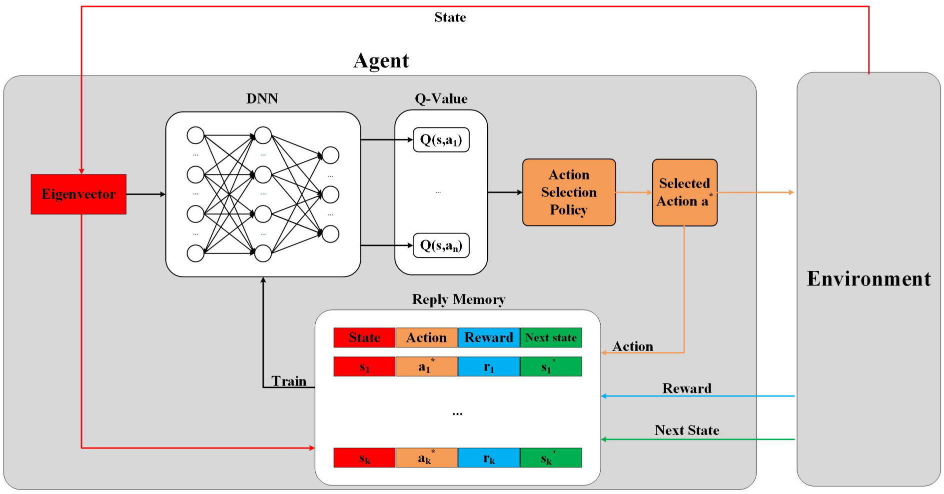

FIGURE 3. This is DQN framework for DRL. DNN outputs the Q values corresponding to all actions according to the feature vector of the input *state*, determines the action *a** that acts on the environment according to the action selection strategy, and the Environment returns the *reward* and the *next state* to the Agent according to the action *a**. The *state*, *action*, *reward*, and *next state* generated during the interaction between the Agent and the Environment are stored in the replay memory, and data is periodically sampled from it for training the DNN.

Environment feedbacks the reward value r^k to the agent according to the action a^* and the current system state and the next state s^{k+1} based on the action a^*, besides r^k will be used to update the Q value table in the agent. The update equation is as follows

$$Q_{\text{update}}\left(s^k, a_t\right) = Q\left(s^k, a_t\right) + \alpha\left[r+\beta \max_{a\in A} Q\left(s^{k+1}, a\right)-Q\left(s^k, a_t\right)\right], \tag{29}$$

where α is the learning rate and β is the attenuation factor. During the interaction between the agent and the environment, the Q value table maintained by the agent is continuously iteratively updated until the Q value in the Q value table converges. The agent can select the action with the greatest value to act on the environment according to the greedy algorithm.

As we mentioned in the second section, at time slot k, the state feature vector of the environment is composed of the following parts: the vehicle speed $v = (v[1], v[2], v[3]\ldots v[M])$, the vehicle position information $p^k = \left(p^k[0], p^k[1], p^k[2]\ldots p^k[M]\right)$, where $p^k[0]$ represents the position of the RSU. The vehicle computing capacity $C = (C[0], C[1], C[2]\ldots C[M])$, where $C[0]$ represents the computing capacity of RSU. The tasks $\left(T^k, d^k\right)$ generated by the vehicle, where $T^k = \left(T^k[1], T^k[2], T^k[3]\ldots T^k[M]\right)$, and $d^k = \left(d^k[1], d^k[2], d^k[3]\ldots d^k[M]\right)$, the cumulative reward value of the vehicle $R^k_{ac} = \left(R^k_{ac}[1], R^k_{ac}[2], R^k_{ac}[3]\ldots R^k_{ac}[M]\right)$. Therefore, the system status can be expressed as

$$s^k = \left\{v, p^k, C, T^k, d^k, R^k_{ac}\right\}, \tag{30}$$

whenever a new system state is generated, the agent will choose a suitable action a^k to act on the environment according to the current system state. The action a^k contains the following

- The amount of computation resources contributed by the vehicle $\xi^k = \left(\xi^k[1], \xi^k[2], \xi^k[3], \ldots \xi^k[M]\right)$, where $\xi^k[i] = j$ represents the computation resources contributed by the i^{th} vehicle The amount is j.
- Offloading decision taken on the task $O^k = \left(O^k[1], O^k[2], O^k[3]\ldots O^k[M]\right)$, where $O^k[i] = j$ means vehicle i offloads its own task to vehicle j for processing.
- Selection of additional transmission bandwidth $W^k = \left(W^k[1], W^k[2], W^k[3]\ldots W^k[M]\right)$, where $W^k[i] = j$ indicates that vehicle i selects additional j bandwidth for task transmission.
- Selection of wireless transmission power $P^k = (P^k[1], P^k[2], P^k[3]\ldots P^k[M])$, where $P^k[i] = j$ indicates that the wireless transmission power of the vehicle i transmission task is j.

Therefore, the action a^k can be expressed as

$$a^k = \left\{\xi^k, O^k, W^k, P^k\right\}. \tag{31}$$

After the action a^k is transmitted to the environment, the environment will generate a reward value r^k and feed it back to the agent. The r^k here can use the results discussed in Section III as

$$r^k = U^{Sk}\left(\xi^k, W^k\right) - \omega_{p^k} T^k P^k, \tag{32}$$

where $\omega_{p^k} T^k P^k$ represents the energy consumed by wireless transmission. The greater the transmission power, the more the energy consumed by wireless transmission, and the smaller the corresponding reward value.

Algorithm 1 Training Deep Q-Network of Computation Resourcement Management

```
Initialize: Replay memory with a size of Ψ,
    the mini-batch with size of ψ, ψ < Ψ, DNN network
    weights θ_DNN, IoT Scenario Environment;
Initialize: k = 0;
while All of vehicles are in the communication range of
    the RSU do
    Input the state of environment s^k;
    Select ε in range (0,1);
    if ε < ϵ then
        Select action a^k randomly;
    else
        Select a^{k*} = arg max_{a∈A} Q(s^k, a^k);
    end
    Environment gengerates next state s^{k+1} and reward
    r^k based on the action of agent;
    Save data items (s^k, a^k, r^k, s^{k+1}) into Replay
    memory;
    if i > ψ then
        Sample mini-batch of data from Replay memory
        randomly;
        Train the DQN with mini-batch data and update
        the DQN weights θ_DNN;
    end
    Update the time index k = k + 1.
end
```

Because the state feature vector space of this scenario increases with the number of vehicles in the environment, the agent needs to maintain a large Q value table, and it takes a lot of iterations to make the Q value converge, so we will use DRL to get convergent Q values faster.

B. DEEP REINFORCEMENT LEARNING

We will use DQN framework for DRL, as shown in Fig. 3.

The Agent contains a DNN. The input of the neural network is the state feature vector s^k of the current environment, and the output of the neural network is the Q value $Q(s, a)$ obtained from all the actions taken under the state s^k. The action a^* is selected to act on the environment through the action selection strategy π. Different from reinforcement learning, on the one hand, DRL introduces a DNN instead of the Q value table, and on the other hand, a reply memory is set up to store parameters $\left(s^k, a^k, r^k, s^{k+1}\right)$ during the interactions between the agent and environment, periodically sampling ω data from the reply memory for training the DNN, and updating the weight parameter θ_{DNN} in the DNN to minimize the loss function. The loss function is as follows

$$\text{Loss}\left(\theta_{DNN}\right) = \sum_{i=1}^{\Omega}\left(y_i^k - Q\left(s_i^k, a_i^k\right)\right)^2, \tag{33}$$

$$y_i^k = r_i^k + \gamma \max_{a\in \text{A}} Q\left(s_i^{k+1}, a\right). \tag{34}$$

In order to be able to traverse the entire state space in this state instead of just choosing the action that maximizes the Q value, during the DNN training, our action selection strategy π is defined as the ϵ-greedy method, ϵ represents the action exploration rate, and ϵ-greedy method indicates that the agent has a probability of $1-\epsilon$ to choose the action a^* that maximizes $Q(s, a)$, and has the probability of ϵ to randomly choose the action a. As the number of iterations increases, the value of ϵ gradually decreases, which accelerates the convergence of the Q value.

In the DQN test process, we choose the strategy π as the greedy method, that is, the action a^* that maximizes the Q value is selected as the output and acts on the environment. The average performance of the feedback is used to measure the system performance level after iteration.

The algorithm for training DQN is shown as in Algorithm 1.

V. NUMERICAL EXPERIMENT

In this section, we will build a DRL simulation environment based on *tensorflow*, and describe the simulation results of system performance.

A. GENERAL SETUP

In the DQN architecture, we set up a four-layer neural network, one of which is an input layer, one is an output layer, and two hidden layers. The number of neurons in the two hidden layers is 600, 300. We will use the ReLu function as the activation function and ReLu function of this neural network is as follows

$$f_{\text{R}}(x) = \max(0, x). \tag{35}$$

At the same time, we will use the Adam optimizer to optimize the Loss function in the neural network [34]. Other detailed parameters about neural networks and default application scenarios [28], [35], which are shown in Table 1.

TABLE 1. Simulation parameters.

Parameter	Value
Carrier frequency	5900MHz
Maximum transmission bandwidth	20MHz
Basic transmission bandwidth	10MHz
Number of lanes	1 in each direction
Coverage radius of RSU communication range	1000m
RSU position coordinate	1000m
Velocity of vehicle	0-20m/s
Time slot interval	1s
Task size	50–150 Mb
Computation resource of vehicle	50–600 MHz
Computation resource of RSU	5 GHz
Latency constraints for task processing	1 s
Task transmit power level	[10,20,30] dBm
Noise power	-114dBm
Vehicle antenna reception gain	5dB
BS antenna reception gain	8dB

According to the setting of the scene and the construction of the model, we will compare the system performance from the following aspects:

- We modify the parameters in the model and compare the differences in the performance of the system with the modification of each parameter to obtain the optimal model parameter settings.
- We compare the resource management model proposed in this paper with conventional resource management solutions. In conventional resource management solutions, when the vehicle's own tasks cannot be processed locally, the tasks are directly offloaded to the RSU for processing, and the system performance of the two solutions is compared.

B. EXPERIMENT RESULT

1) EXPERIMENT 1

We implemented experimental simulations under different parameters, such as the number of vehicles, the size of mini-batch, the size of the reply memory, and the learning rate, and compared and analyzed the simulation.

In actual application scenarios, the number of vehicles will affect the number of samples obtained by the system at the end of each episode. The larger the number of vehicles, the more samples the system obtains after each episode, and the earlier the optimization of the neural network and the parameter update can be performed. As shown in Fig. 4, it is a comparison of system performance under different vehicle numbers [25, 50, 100, 150] in the same scene. The ordinate represents the standardized system performance, which is represented by the system delay, energy consumption, task completion rate, and vehicle contribution resources. Through the simulation graph, we can know that when the number of vehicles in the environment is veh num $=$ 25, the performance curve starts to rise around episode $=$ 10. As the number of vehicles in the system increases, the performance curve improves faster. When the number of vehicles veh num $=$ 100, 150, the performance curve converges at the same time. Therefore, when there are more vehicles in the environment, it is more beneficial to the optimization of the neural network in DQN.

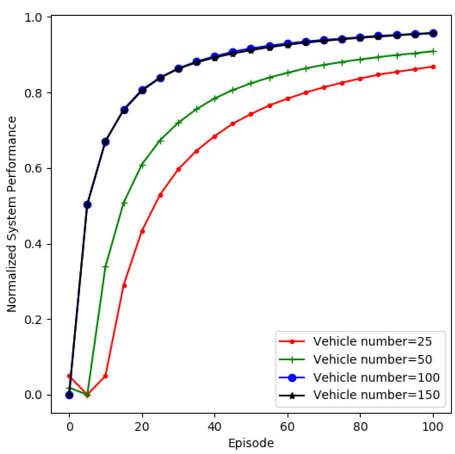

FIGURE 4. **The normalized system performance versus different numbers of vehicles.**

As mentioned above, when there are more vehicles in the system, more sample values can be obtained in an episode. In our proposed model, when reply memory $>$ mini-batch, the neural network starts optimization. Therefore, we infer that when the number of vehicles is fixed, as the number of samples increases, the system performance will start to converge later. We assume that the number of vehicles in the current system is veh num $=$ 25, and the number of samples is mini-batch $=$ [50, 100, 200, 300]. Fig. 5 shows our experimental simulation results, which is consistent with our inference, that is, as the mini-batch increases, the later the system performance improves.

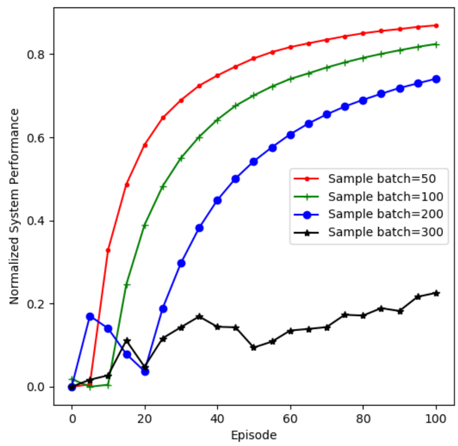

FIGURE 5. **The normalized system performance versus different sizes of mini-batch.**

The size of the reply memory has been set in the model. If the data stored in the reply memory is full, the older data will be cleared to make room for new data to store. We infer that if the size of the reply memory becomes larger, there will be more old data in sample data, and it is less likely that new data is sampled. In other words, the new data cannot be used to train the neural network in time, delaying the improvement of system performance. We assume that the current system has veh num $=$ 25, sample number mini-batch $=$ 50, reply memory $=$ [500, 1000, 2000, 3000]. The simulation results are shown in Fig. 6. When reply memory $=$ [500, 1000] The system performance is basically improved at the same time. When reply-memory $=$ [2000, 3000], the system performance is relatively late.

We hope that the system model can pay attention to long-term benefits rather than immediate benefits, which is reflected in the gamma value in equation (34). A large gamma value indicates that there is a large part of the value from the next state in current Q-value, which means that the model will consider more about the reward the next state will obtain. As shown in Fig. 7, for a system model with a large gamma value, its performance can also be quickly improved.

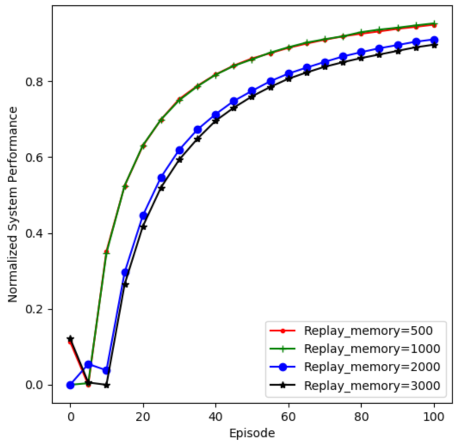

FIGURE 6. The normalized system performance versus different sizes of reply memory.

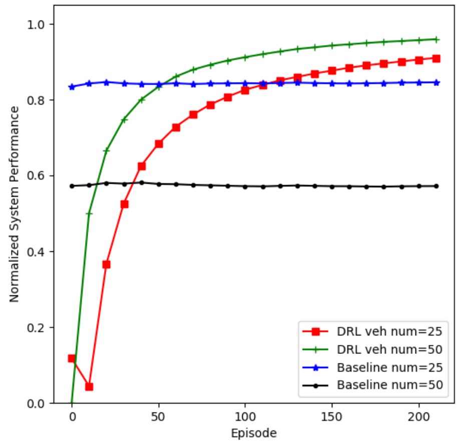

FIGURE 8. The normalized system performance versus different statrgies of resource management.

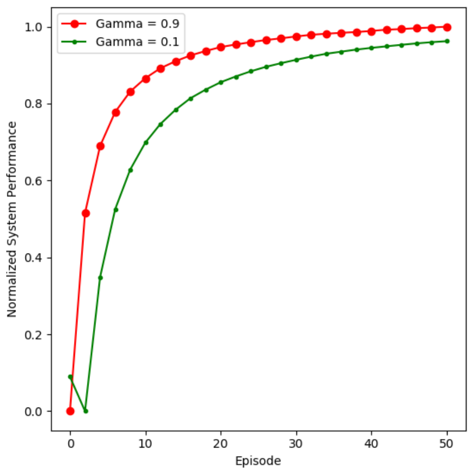

FIGURE 7. The normalized system performance versus different values of gamma.

2) EXPERIMENT 2

We compare the performance of the proposed system model with the conventional model. In the conventional model, when the vehicle cannot complete the task within the maximum delay of the task, it will directly offload the task to the RSU for processing. Although the RSU has relatively strong computing power, it can smoothly handle all tasks offloaded to the RSU in a scenario with a small number of vehicles. However, as the number of vehicles increases, the number of tasks offloaded to the RSU gradually increases, and the task processing pressure on the RSU increases too. As the RSU is in a full load state, the RSU has no free computation resources for other offloaded tasks to utilize. Hence, these tasks will not be processed successfully. Therefore, the larger the number of vehicles, the higher the failure rate of task processing, and the lower the performance level of the conventional offloading policy. As shown in Fig. 8, with the increase in the number of vehicles, the task offloading scheme proposed in this paper can achieve performance improvement earlier, and the system performance has not received much impact, because with the increase in the number of vehicles, the number of tasks that vehicles offload to the RSU increases, but at the same time, the number of vehicles with idle computation resources also increases, so the computation resources contributed by vehicles also increase. However, the performance of the conventional offloading method in Fig. 8 decreases as the number of vehicles increases. Therefore, the resource management scheme proposed in this paper can motivate vehicles to contribute resources, and it is robust to the increase in the number of vehicles. As the number of vehicles increases, the system can maintain good performance.

VI. CONCLUSION

In this article, we first designed a resource management scheme based on contract theory. The contract theory contains two aspects of the contract. One is a resource contribution contract. The vehicle contributes resources and obtains a reward value corresponding to the amount of resources it contributes, which motivates vehicles to contribute computation resources to the RSU. The other is a contract for resource utilization. The vehicle uses its accumulated reward in exchange for additional wireless transmission bandwidth to improve the QoS of the vehicle application. For the entire system, while improving the overall performance of the system (delay, task completion rate, energy consumption), it is also necessary to minimize the utilization of system resources (spectrum resources). We use this as a basis to establish an optimization function for system performance.

Secondly, we use the DRL method to implement the proposed contract-based resource management and task offloading scheme. We use the previously established system performance optimization function as the reward in reinforcement learning. We set a reply memory, and use the data sampled from the reply memory to update the parameters in the deep neural network to optimize resource management policy and the decision of tasks offloading and improves system performance.

VII. FUTURE WORK

In the DRL model proposed in this paper, DQN outputs the Q values of all actions corresponding to the state feature vector. With the increase of the number of vehicles, the space size of the action a for representing the offloading decision and resource management will be very large, so the output value space of the DQN network will be very large, which will affect the implementation effect of the action selection strategy.

Therefore, in future work, we can use other DRL methods to improve the efficiency of the algorithm, such as Deep Deterministic Policy Gradient (DDPG). This method is a policy-based DRL algorithm, that is, in DDPG, a Action network can directly generate corresponding output actions based on the input state vector. A Critic network in DDPG will evaluate the actions generated by the Action network and continuously optimize the action selection strategy of the Action network [36]–[38].

REFERENCES

[1] X. Ma, J. Zhao, Y. Gong, and X. Sun, "Carrier sense multiple access with collision avoidance-aware connectivity quality of downlink broadcast in vehicular relay networks," *IET Microw., Antennas Propag.*, vol. 13, no. 8, pp. 1096–1103, Jul. 2019.

[2] Q. Li, J. Zhao, and Y. Gong, "Computation offloading and resource allocation for mobile edge computing with multiple access points," *IET Commun.*, vol. 13, no. 17, pp. 2668–2677, Oct. 2019.

[3] J. Zhao, Q. Li, Y. Gong, and K. Zhang, "Computation offloading and resource allocation for cloud assisted mobile edge computing in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 7944–7956, Aug. 2019.

[4] A. A. Alahmadi, A. Q. Lawey, T. E. H. El-Gorashi, and J. M. H. Elmirghani, "Distributed processing in vehicular cloud networks," in *Proc. 8th Int. Conf. Netw. Future (NOF)*, London, U.K., Nov. 2017, pp. 22–26.

[5] M. Nabi, R. Benkoczi, S. Abdelhamid, and H. S. Hassanein, "Resource assignment in vehicular clouds," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Paris, France, May 2017, pp. 1–6.

[6] X. Wei, S. Wang, A. Zhou, J. Xu, S. Su, S. Kumar, and F. Yang, "MVR: An architecture for computation offloading in mobile edge computing," in *Proc. IEEE Int. Conf. Edge Comput. (EDGE)*, vol. 27, Sep. 2017, pp. 89–95.

[7] Q. Li, J. Zhao, Y. Gong, and Q. Zhang, "Energy-efficient computation offloading and resource allocation in fog computing for Internet of everything," *China Commun.*, vol. 16, no. 3, pp. 32–41, Mar. 2019.

[8] J. Zhang, W. Xia, F. Yan, and L. Shen, "Joint computation offloading and resource allocation optimization in heterogeneous networks with mobile edge computing," *IEEE Access*, vol. 6, pp. 19324–19337, 2018.

[9] C. You, K. Huang, H. Chae, and B.-H. Kim, "Energy–efficient resource allocation for mobile–edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1397–1411, Mar. 2017.

[10] L. Huang, S. Bi, and Y.-J. A. Zhang, "Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks," 2018, *arXiv:1808.01977*. [Online]. Available: https://arxiv.org/abs/1808.01977

[11] P. Zhao, H. Tian, C. Qin, and G. Nie, "Energy–saving offloading by jointly allocating radio and computational resources for mobile edge computing," *IEEE Access*, vol. 5, pp. 11255–11268, 2017.

[12] Z. Ning, J. Huang, X. Wang, J. J. P. C. Rodrigues, and L. Guo, "Mobile edge computing–enabled Internet of vehicles: Toward energy–efficient scheduling," *IEEE Netw.*, vol. 33, no. 5, pp. 198–205, Sep. 2019.

[13] Z. Ning, J. Huang, and X. Wang, "Vehicular fog computing: Enabling real–time traffic management for smart cities," *IEEE Wireless Commun.*, vol. 26, no. 1, pp. 87–93, Feb. 2019.

[14] X. Wang, Z. Ning, and L. Wang, "Offloading in Internet of vehicles: A fog–enabled real–time traffic management system," *IEEE Trans. Ind. Informat.*, vol. 14, no. 10, pp. 4568–4578, Oct. 2018.

[15] X. Wang, Z. Ning, X. Hu, L. Wang, L. Guo, B. Hu, and X. Wu, "Future communications and energy management in the Internet of vehicles: Toward intelligent energy–harvesting," *IEEE Wireless Commun.*, vol. 26, no. 6, pp. 87–93, Dec. 2019, doi: 10.1109/mwc.001.1900009.

[16] Y. Lai, F. Yang, L. Zhang, and Z. Lin, "Distributed public vehicle system based on fog nodes and vehicular sensing," *IEEE Access*, vol. 6, pp. 22011–22024, 2018.

[17] Q. Yuan, H. Zhou, J. Li, Z. Liu, F. Yang, and X. S. Shen, "Toward efficient content delivery for automated driving services: An edge computing solution," *IEEE Netw.*, vol. 32, no. 1, pp. 80–86, Jan. 2018.

[18] H. Zhang, Q. Zhang, and X. Du, "Toward vehicle–assisted cloud computing for smartphones," *IEEE Trans. Veh. Technol.*, vol. 64, no. 12, pp. 5610–5618, Dec. 2015.

[19] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular fog computing: A viewpoint of vehicles as the infrastructures," *IEEE Trans. Veh. Technol.*, vol. 65, no. 6, pp. 3860–3873, Jun. 2016.

[20] H. Peng, D. Li, Q. Ye, K. Abboud, H. Zhao, W. Zhuang, and X. Shen, "Resource allocation for cellular-based inter-vehicle communications in autonomous multiplatoons," *IEEE Trans. Veh. Technol.*, vol. 66, no. 12, pp. 11249–11263, Dec. 2017.

[21] Z. Zhou, P. Liu, J. Feng, Y. Zhang, S. Mumtaz, and J. Rodriguez, "Computation resource allocation and task assignment optimization in vehicular fog computing: A contract–matching approach," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3113–3125, Apr. 2019.

[22] Y. Zhang, M. Pan, L. Song, Z. Dawy, and Z. Han, "A survey of contract theory–based incentive mechanism design in wireless networks," *IEEE Wireless Commun.*, vol. 24, no. 3, pp. 80–85, Jun. 2017.

[23] Y. Zhang, L. Song, W. Saad, Z. Dawy, and Z. Han, "Contract–based incentive mechanisms for device-to-device communications in cellular networks," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 10, pp. 2144–2155, Oct. 2015.

[24] H. Mao, M. Alizadeh, I. Menache, and S. Kandula, "Resource management with deep reinforcement learning," in *Proc. 15th ACM Workshop Hot Topics Netw. HotNets*, Nov. 2016, pp. 50–56.

[25] Z. Ning, Y. Li, P. Dong, X. Wang, M. S. Obaidat, X. Hu, L. Guo, Y. Guo, J. Huang, and B. Hu, "When deep reinforcement learning meets 5G vehicular networks: A distributed offloading framework for traffic big data," *IEEE Trans. Ind. Informat.*, to be published, doi: 10.1109/tii.2019.2937079.

[26] Z. Ning, Y. Feng, M. Collotta, X. Kong, X. Wang, L. Guo, X. Hu, and B. Hu, "Deep learning in edge of vehicles: Exploring trirelationship for data transmission," *IEEE Trans. Ind. Informat.*, vol. 15, no. 10, pp. 5737–5746, Oct. 2019.

[27] Z. Ning, P. Dong, X. X. Wang, and J. Rodrigues, "Deep reinforcement learning for vehicular edge computing: An intelligent offloading system," *ACM Trans. Intell. Syst. Technol.*, vol. 25, p. 1, May 2019.

[28] Z. Xu, Y. Wang, J. Tang, J. Wang, and M. C. Gursoy, "A deep reinforcement learning based framework for power-efficient resource allocation in cloud RANs," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6.

[29] M. A. Salahuddin, A. Al-Fuqaha, and M. Guizani, "Reinforcement learning for resource provisioning in the vehicular cloud," *IEEE Wireless Commun.*, vol. 23, no. 4, pp. 128–135, Aug. 2016.

[30] Y. He, N. Zhao, and H. Yin, "Integrated networking, caching, and computing for connected vehicles: A deep reinforcement learning approach," *IEEE Trans. Veh. Technol.*, vol. 67, no. 1, pp. 44–55, Jan. 2018.

[31] J. Zhao, X. Guan, and X. P. Li, "Power allocation based on genetic simulated annealing algorithm in cognitive radio networks," *Chin. J. Electron.*, vol. 22, no. 1, pp. 177–180, Jan. 2013.

[32] J. Zhao, S. Ni, L. Yang, Z. Zhang, Y. Gong, and X. You, "Multiband cooperation for 5G HetNets: A promising network paradigm," *IEEE Veh. Technol. Mag.*, vol. 14, no. 4, pp. 85–93, Dec. 2019.

[33] Z. Junhui, Y. Tao, G. Yi, W. Jiao, and F. Lei, "Power control algorithm of cognitive radio based on non-cooperative game theory," *China Commun.*, vol. 10, no. 11, pp. 143–154, Nov. 2013.

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

[35] X. Chen, H. Zhang, C. Wu, S. Mao, Y. Ji, and M. Bennis, "Optimized computation offloading performance in virtual edge computing systems via deep reinforcement learning," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4005–4018, Jun. 2019.

[36] C. Qiu, Y. Hu, Y. Chen, and B. Zeng, "Deep deterministic policy gradient (DDPG)–based energy harvesting wireless communications," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8577–8588, Oct. 2019.
[37] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," in *Proc. ICLR*, 2016.
[38] S.-C. Tseng, Z.-W. Liu, Y.-C. Chou, and C.-W. Huang, "Radio resource scheduling for 5G NR via deep deterministic policy gradient," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, May 2019, pp. 1–6.

JUNHUI ZHAO (Senior Member, IEEE) received the M.S. and Ph.D. degrees from Southeast University, Nanjing, China, in 1998 and 2004, respectively. From 1998 to 1999, he was with the Nanjing Institute of Engineers, ZTE Corporation. Then, he was an Assistant Professor with the Faculty of Information Technology, Macao University of Science and Technology, in 2004, and continued there as an Associate Professor, until 2007. In 2008, he joined Beijing Jiaotong University as an Associate Professor, where he is currently a Professor with the School of Electronics and Information Engineering. Meanwhile, he was also a short term Visiting Scholar with Yonsei University, South Korea, in 2004, and a Visiting Scholar with Nanyang Technological University, Singapore, from 2013 to 2014. Since 2016, he has been with the School of Information Engineering, East China Jiaotong University. His current research interests include wireless and mobile communications and related applications that contain 5G mobile communication technology, high-speed railway communications, vehicle communication networks, wireless localization, and cognitive radios.

MING KONG received the B.Eng. degree in electronic science and technology from Beijing Jiaotong University, Beijing, China, in 2016, where he is currently pursuing the M.S. degree. His research interests include 5G vehicular networks and resource allocation.

QIUPING LI received the B.Sc. degree in electronic information science and technology from China West Normal University, China, in 2015. She is currently pursuing the Ph.D. degree in communication and information systems from Beijing Jiaotong University, Beijing, China. Her research interests include heterogeneous cellular networks, vehicular networks, mobile edge computing, resource allocation, and optimization techniques.

XIAOKE SUN received the B.E. degree in electronic information science and technology from Xiangtan University, China, in 2016. She is currently pursuing the Ph.D. degree in communication and information systems from Beijing Jiaotong University, Beijing, China. Her research interests include 5G vehicular networks, mobile edge computing, resource allocation, and stochastic optimization.