IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 21, 2019, accepted December 7, 2019, date of publication December 13, 2019,
date of current version December 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2959348

Joint Optimization of Data Offloading and Resource Allocation With Renewable Energy Aware for IoT Devices: A Deep Reinforcement Learning Approach

HONGCHANG KE[1,2,3], JIAN WANG[1,3], HUI WANG[4], AND YUMING GE[5]

[1]College of Computer Science and Technology, Jilin University, Changchun 130012, China
[2]School of Computer Technology and Engineering, Changchun Institute of Technology, Changchun 130012, China
[3]Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China
[4]College of Computer Science and Engineering, Changchun University of Technology, Changchun 130012, China
[5]Technology and Standards Research Institute, China Academy of Information and Communications Technology, Beijing 100191, China

Corresponding author: Jian Wang (wangjian591@jlu.edu.cn)

This work was supported in part by the National Nature Science Foundation under Grant 61572229 and Grant 6171101066, in part by the Jilin Provincial Science and Technology Development Foundation under Grant 20170204074GX, Grant 20180201068GX, and Grant 20190302106GX, in part by the Jilin Provincial International Cooperation Foundation Grant 20180414015GH, in part by the CERNET Innovation Project under Grant NGII20170413, and in part by the Scientific and Technological Planning Project of Jilin Province under Grant 2018C036-1.

ABSTRACT A large number of connected sensors and devices in Internet of Things (IoT) can generate large amounts of computing data and increase massive energy consumption. Real-time states monitoring and data processing of IoT nodes are of great significance, but the processing power of IoT devices is limited. Using the emerging mobile edge computing (MEC), IoT devices can offload computing tasks to an MEC server associated with small or macro base stations. Moreover, the use of renewable energy harvesting capabilities in base stations or IoT nodes may reduce energy consumption. As wireless channel conditions vary with time and the arrival rates of renewable energy, computing tasks are stochastic, and data offloading and renewable energy aware for IoT devices under a dynamic and unknown environment are major challenges. In this work, we design a data offloading and renewable energy aware model considering an MEC server performing multiple stochastic computing tasks and involving time-varied wireless channels. To optimize data transmission delay, energy consumption, and bandwidth allocation jointly, and to avoid the curse of dimensionality caused by the complexity of the action space, we propose a joint optimization method for data offloading, renewable energy aware, and bandwidth allocation for IoT devices based on deep reinforcement learning (JODRBRL), which can handle the continuous action space. JODRBRL can minimize the total system cost(including data buffer delay cost, energy consumption cost, and bandwidth cost) and obtain an efficient solution by adaptively learning from the dynamic IoT environment. The numerical results demonstrate that JODRBRL can effectively learn the optimal policy, which outperforms Dueling DQN, Double DQN (DDQN), and greedy policy in stochastic environments.

INDEX TERMS Data offloading, energy aware, mobile edge computing, deep reinforcement learning.

I. INTRODUCTION

IoT devices are widely used in various fields, such as industrial control, network equipment systems, public safety equipment, and environmental monitoring [1]. IoT devices connect to the cellular network and have diverse capabilities, such as storage, computing, and communication and most IoT nodes in current applications such as narrowband IoT are suitable for delay-insensitive tasks, such as smart electricity measuring and fleet management [2]. Some IoT devices need only periodically transmit data to the central processor. However, this does not apply to more sophisticated IoT devices, such as telemedicine and autonomous driving, which are delay-sensitive and require high-reliability communications

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

with low delay [3], [4]. Therefore, data integrity, low latency, and energy saving are currently required for IoT devices. However, the increase in the amount of task data will inevitably lead to enormous energy consumption. The gradual development of heterogeneous IoT has brought challenges that should be resolved [5].

MEC has been proposed as a solution that enables wireless IoT devices (e.g., mobile phones and smart watches) to offload computing tasks to MEC servers according to the base station which can significantly improve computing efficiency and reduce processing delays and energy consumption [6]. MEC can improve the quality of experience (QoE), save bandwidth resources, and sink computing tasks to mobile edge nodes [7]. MEC can resolve the contradiction between the limited resources on mobile devices and the growing computing needs of IoT nodes [8]. However, the offloading of IoT computing tasks to the MEC server requires fast and efficient wireless data transmission [9]. For instance, wireless communication resources for computing offloading should be properly allocated which can meet the computing and transmission delay requirements. Furthermore, MEC is usually deployed in 5G ultra-dense cellular networks with small base stations and macro base stations which may cause massive energy consumption. The Technology about network slicing is a key enabler for RAN sharing, which can improve the QoS of MEC. However, few works have been discussed about network slicing because of the complex structure of MEC.

The use of renewable energy to power small base stations is a feasible solution for reducing energy consumption. Energy harvesting (EH) is another promising technique that increases the quality of service (QoS) for small base stations and IoT devices [10], [11]. Energy storage equipment with the EH module enables the base station to harvest green energy, such as solar radiation and wind power, so that the energy consumption problem may be alleviated.

IoT devices can offload intensive tasks to a MEC server and use the trade-off between communication and computing to realize energy savings and performance enhancements [12], [13]. Currently, most works are concerned with computing tasks by the binary offloading method, whereby each IoT device selects whether to execute the task locally or offload it to the MEC server [14], [15]. In the heterogeneous network environment of massive IoT nodes, it is more reasonable to use partial local execution and partial offloading to the MEC server. Moreover, some studies focus on the EH of wireless terminals, but the wireless communication environment will cause energy loss, and when the number of IoT nodes increases, energy management becomes difficult.

Under time-varying channel state and renewable energy supply, we propose a joint optimization scheme for transmission delay, renewable energy consumption, and bandwidth allocation based on deep reinforcement learning (JODRBRL). The main contributions of this study are summarized as follows:

- We first design an MEC framework in a heterogeneous cellular network that includes multiple IoT nodes, multiple energy storage and processing units, and an MEC server. IoT devices can communicate with a small base station with a high-performance processor and a macro base station with a MEC server. The aim of the proposed MEC framework is to minimize the total system cost of energy consumption, data transmission delay, and bandwidth allocation under time-varying channel state.
- We use renewable energy to power the small base stations by energy access point(AP) with energy storage equipment in the proposed MEC model and IoT device equipped with an EH module can harvest energy from the energy AP. According to the energy supply of the small base station, the computing task can be partially performed by a local processor or partially offloaded to the MEC server.
- We propose a joint optimization scheme for data transmission delay, energy consumption, and bandwidth allocation based on deep deterministic policy gradient (DDPG), which is a model-free deep-reinforcement learning (DRL) method and can efficiently handle continuous action space. For more effective stochastic exploration, we add the OU noise vector to the action space. In terms of computing tasks, JODRBRL can adaptively perform local execution and offload tasks to the small base station or the MEC server. Under the conditions of time-varying wireless channel state and bandwidth, JODRBRL can learn the optimal policy in the joint optimization scheme.
- We perform extensive numerical simulations under time-varying channel state and available bandwidth in the proposed MEC system to verify the effectiveness of JODRBRL. The results demonstrate that the performance of JODRBRL surpasses that of three benchmark algorithms.

The reminder of the paper is organized as follows. Section II reviews related works. Section III presents the MEC model and describes in detail the communication, energy, bandwidth allocation, and computation models. The problem statement and the description of DRL for JODRBRL are presented in Section IV. The results of the numerical simulations are discussed in Section V. Finally, Section VI concludes the paper.

II. RELATED WORK

In terms of related works on the renewable energy supply and EH, Liu et al. defined an energy consumption model which consisted of a solar energy collection and a battery for energy storage. The energy data was used by the National Solar Radiation Database at DENVER/CENTENNIAL and a security disjoint routing-based verified message(SDRVM) method was proposed which significantly reduced the energy consumption of the wireless nodes [16]. Bi et al. presented a wireless powered communication(WPC) model and a radio frequency(RF) energy receiver model. The proposed model

described the communicating processing between the wireless devices(WDs) with EH and the energy transmitting devices [17]. Zhou *et al.* applied a power minimization scheme to the constraints on the number of computation bits and energy harvesting causality [18]. Min *et al.* presented a learning-based computing offloading scheme for IoT devices with EH to maximize the system utility, including energy consumption and computation latency [19].

A number of works had been concerned with optimization for data transmission delay and energy consumption in wireless networks. Liu et al. proposed an adjusting forwarder nodes and duty cycle using packet aggregation routing (AFNDCAR) scheme to decrease a large number of redundant data and transmission delay in the body sensor networks(BSNs). The proposed algorithm flexibly used duty cycle to control the residual energy of nodes which improved the efficiency of delay and energy consumption [20]. Bi *et al.* formulated a joint optimization method for local computing or offloading to an MEC server and transmission time allocation. The scheme was based on the alternating direction method of multipliers (ADMM) to maximize the computation rate [21]. Considering task offloading among vehicles in heterogeneous vehicular networks, Sun et al. designed an adaptive learning-driven task offloading algorithm based on the multi-armed bandit algorithm to reduce the average offloading delay [22]. Lyu *et al.* proposed an asymptotically optimal method for the tradeoff between offloading and local execution. The method ensured the time complexity of task delays and the maximum energy saving at a $[O(\epsilon), O(1/\epsilon)]$-tradeoff [23].

For computing task offloading and resource allocation in heterogeneous wireless networks. Zhang et al. studied an optimization problem to minimize the energy consumption of an MEC offloading system in 5G heterogeneous networks and jointly optimized the computation offloading decisions and the radio resource allocation strategies by a three-stage energy-efficient method [24]. Zhou *et al.* used queuing theory to derive stochastic traffic models and proposed an ADMM-based energy-efficient resource allocation algorithm in vehicular networks. The proposed method transformed an NP-hard problem to a convex global consensus problem [25]. Lyu *et al.* proposed submodular set function optimization for the joint optimization of the offloading decision and the communication and computation resource allocation to maximize the system utility [26].

In recent years, some works had been based on deep learning (DL) and DRL to jointly optimize computing offloading and resource allocation or transmission delay and energy consumption. Xu et al. introduced a joint offloading and edge server provisioning problem based on RL to minimize the long-term system cost and used a post-decision state(PDS) learning algorithm to implement the special structure of state transitions in MEC system [27]. Wei *et al.* introduced a joint optimization solution for caching, computing, and communication, and used the actor-critic RL method, which can learn the optimal policy for minimizing the data transmitting latency [28]. Quan *et al.* developed a two-layered RL algorithm to implement task offloading to an MEC server under limited resources for the mobile devices, thus obtaining a tradeoff between the utilization rate of the physical machine and delay [29]. Li *et al.* proposed an online RL scheme for load balancing in vehicular networks. The scheme took advantage of historical association data to maximize vehicle service rates [30]. Hu *et al.* designed a DRL-based offloading model and used a three-layer neural network to learn the optimal offloading policy with different data transmission rates [31]. Ning *et al.* modelled the architecture of communication and edge computing and proposed a DRL-based joint optimization scheme for computing task scheduling and wireless resource allocation in vehicular networks [32].

In summary, existing works considered binary offloading for computing tasks and either introduced the joint optimization scheme for caching and computing or minimized data delay and energy consumption under the channel constraints. For RL optimization, most works used the scheme with discrete action space, such as DQN and DDQN. Discretization of the action space could affect the expected cumulative rewards of DRL and further influence the optimization results. Considering the time-varying channel state as well as stochastic computing tasks and renewable energy supply, we proposed a joint optimization scheme for data delay, energy consumption, and bandwidth allocation based on deep reinforcement learning with continuous action space which could minimize the total cost including buffer delay cost, energy consumption cost and bandwidth cost.

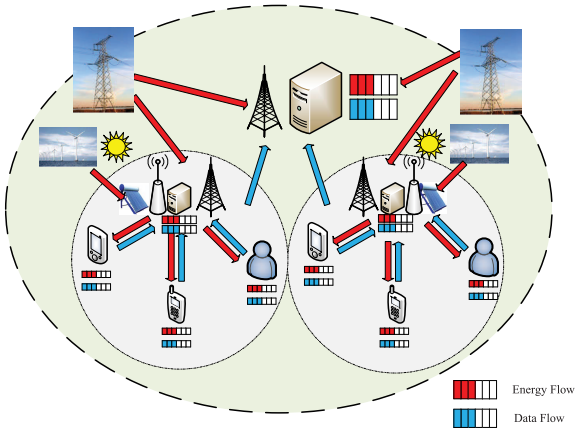


FIGURE 1. MEC system model with renewable energy supply.

III. SYSTEM MODEL

We design an MEC system with renewable energy supply. It includes multiple IoT nodes, multiple energy storage and processing units, and an MEC server. As shown in Fig. 1, The system consists of three parts: IoT nodes layer, the service equipment layer and the MEC layer. IoT nodes are composed of smart phone, visual terminal and so on. The service equipment layer include a small base station, an energy AP with storage and harvest function and a high-performance processor. The MEC layer consists of an MEC server for

computing and a macro base station. In terms of energy flow, The IoT devices are covered by a small base station. The AP in service equipment layer can continuously harvest and store renewable energy until the store space is full. The AP can power small base station and the IoT devices with EH. When the renewable energy is insufficient, small base station, the IoT devices and MEC server are powered by conventional power grid. For data flow, the high-performance processor connected to small base station can process the computing tasks offloaded by IoT devices, which are less powerful than the MEC server. However, the data generated by IoT devices can be partially executed locally or partially offloaded to the small base station with the high-performance processor or the MEC server by macro base station. Computing offloading to the MEC server can be performed centrally by the small base stations. Let $\mathcal{N} = \{1, 2, \cdots, N\}$ denote the set of IoT devices. In the service equipment layer, each set of equipment includes an energy AP and a small base station with a high-performance processor. Let $\mathcal{K} = \{1, 2, \cdots, K\}$ be the equipment set. Then, there are K energy APs and K small base stations in the MEC model. We assume that the IoT node n has computing task to be processed within the coverage of the small base station k, where $n \in \mathcal{N}$ and $k \in \mathcal{K}$. The IoT device n can harvest the energy from the energy AP k in the covered region as long as the energy queue is not empty. In the MEC layer, there are an MEC server and a macro base station. The macro base station covers all IoT devices and small base stations in the system. However, as the wireless communication region of a small base station is limited, the IoT devices in the k-th covered region can communicate only with the small base station k and select whether to offload the tasks or not. For convenience, by data offloading to the small base station or MEC server, we mean data offloading to the high-performance processor by a small base station or to the MEC server by a macro base station, respectively. Table 1 lists the main notations (and their definitions) used in this work.

TABLE 1. List of main notations.

Notation	Definition
$\mathcal{T}$	Index set of the time slots
$\mathcal{N}$	Set of IoT devices
$\mathcal{K}$	Set of small base stations or energy APs
$H_{m,k}(t)$	Channel vector between the small base station k and the macro base station m in the time slot t
$h_{k,n}(t)$	Total channel gain between the IoT device n and the small base station k in the time slot t
$P_{l,n}(t)$	Allocated power of IoT device n for local execution in the time slot t
$P_{k,n}(t)$	Transmission power of the IoT device n for offloading to the small base station k in the time slot t
$P_{m,k}(t)$	Transmission power of the small base station k for offloading to MEC server m in the time slot t
$A_n^{(D)}(t)$	Size of computing tasks requested by the IoT device n in the time slot t
$A_k^{(E)}(t)$	Size of renewable energy harvested by the energy AP k in the time slot t
ζ_k	SINR of the small base station k
τ_k	Allocated bandwidth for the small base station k
τ_n	Allocated bandwidth for the IoT device n
$\lambda_n^{(D)}$	Arrival rate of computing tasks
$\lambda_k^{(E)}$	Arrival rate of renewable energy

A. COMMUNICATION MODEL

We assume that the channel state between each small base station and the macro base station in the MEC model is varies with time, that is, the channel state is different in each time slot. In our proposed MEC system, the computing tasks to be processed are computation-intensive. So the IoT devices process the computing tasks while harvest energy to ensure that the tasks can be completed. We divide the sequential process time into discrete time slots (or epochs) and each time slot t is a constant. We denote the index set of time slots as $\mathcal{T} = \{1, 2, \cdots, T\}$. In our work, the time slot t is set to 2 ms and because the computing speed of MEC and high-performance processor is so fast that we ignore processing time and backhaul time for computing offloading.

In the designed MEC system, the macro base station includes $\widetilde{N}$ antennas. Thus, there are K small base stations in the MEC system, where $\widetilde{N} > K$. The macro base station does not communicate directly with the IoT nodes but rather with the small base stations. We assume that the uplink communication mode of the MEC system is MIMO. Then, the channel state vector in the time slot t is

$$H_{m,k}(t) = \left[h_{m,k}^{(1)}, h_{m,k}^{(2)}, \cdot, h_{m,k}^{(\widetilde{n})}\right] \tag{1}$$

where the channel vector $H_{m,k}(t)$ undergoes Rayleigh fading. $\widetilde{n}$ represents the number of busy antennas in the macro base station. We use the time correlation autoregressive model [33], [34] to represent the transformation of the channel state between time slots $t - t'$ and t:

$$H_{m,k}(t) = \rho_c H_{m,k}(t - t') + \sqrt{1 - \rho_c^2} e_c(t) \tag{2}$$

where ρ_c is the normalized correlation coefficient between $t - t'$ and t. ρ_c is close to 1. $e_c(t)$ is the error vector and $e_c(t) \sim CN(0, I_k)$, which is the complex Gaussian distribution and is uncorrelated with $H_{m,k}(t)$.

Let H be the $M \times K$ channel matrix and A be the zero-forcing (ZF) linear detector matrix that is related to H [33], [34]. According to the ZF linear detector, the received vector of the channel can be written as

$$r_{zf}(t) = \sqrt{P_{m,k}(t)} A^H(t) H(t) x(t) + A^H(t) n(t) \tag{3}$$

where $A(t) = H(t)\left(H^H(t)H(t)\right)^{-1}$ for ZF, $P_{m,k}(t)$ is the transmitting power of equipment set k for offloading to the MEC server, $x(t)$ is the data symbol, and $n(t)$ is a vector of additive white Gaussian noise with zero mean.

We do not consider the interference among different IoT devices in the same equipment sets in the service equipment layer. However, the IoT nodes between different equipment sets can cause interference. The signal-to-interference-plus-noise ratio (SINR) of equipment set k in time slot t can be written as

$$\zeta_k(t) = \frac{P_{m,k}(t) \cdot \left|a_k^H(t) h_{m,k}(t)\right|^2}{\sum_{j \neq k, j \in \mathcal{K}} \left[P_{m,j}(t) \cdot \left|a_k^H(t) h_{m,j}(t)\right|^2 + \|a_k(t)\|^2\right]} \tag{4}$$

where a_k and $h_{m,k}$ are the k-th columns of the matrices A and H, respectively.

B. ENERGY MODEL

In the designed MEC model, according to [16], [17], each small base station is equipped with an energy AP with EH components (e.g., solar panels and/or wind turbines) to store renewable energy. The energy AP can allocate the harvested energy to the IoT devices within the covered region. In addition to the renewable energy, the small base station can obtain conventional energy from the power grid. The IoT devices should use renewable energy as much as possible to reduce the cost of energy consumption. In this section, we present the energy consumption model when IoT devices perform local execution or computing offload to the MEC server.

Even though renewable energy can reduce energy cost in the system model and enhance the sustainability of IoT devices, the utilization of renewable energy may be affected by environmental factors. The arrival of renewable energy is stochastic. We assume that the arrival rate of renewable energy follows the Poisson distribution with $\lambda_k^{(E)}$ at time slot t; moreover, the arrival of renewable energy in each time slot t is assumed to be independent and identically distributed (i.i.d.), as a small base station can continue to work when there is sufficient energy supply, that is, $\mathbb{E}\left[A_k^{(E)}(t)\right] = \lambda_k^{(E)}$.

According to the EARTH project [35], the power consumption of the small station k can be modeled by the static baseline power and the allocated power:

$$E_k(t) = \left[P_{s,k}(t) + \eta P_k(t)\right] * t \tag{5}$$

where $P_{s,k}(t)$ denotes the static power offset (including, e.g., the baseband processor and the cooling system), the coefficient η is the inverse of the power amplifier efficiency factor, and $P_k(t)$ is the total consumed wireless transmission power of the small base station k. Then, in time slot t, the total consumed power $p_k(t)$ of small base station k can be written by

$$P_k(t) = \sum_{n=1}^{N} P_{l,n}(t) + \xi_k \sum_{n=1}^{N} P_{k,n}(t) + (1-\xi_k) P_{m,k}(t) \tag{6}$$

where $P_{l,n}(t)$ is the consumed power of IoT device covered by small base station k for local execution. $P_{n,k}(t)$ denotes the transmit power between the IoT device n and the small base station k. $P_{m,k}$ is the transmit power between the small base station k and the MEC server. ξ_k is the ratio of offloading to small base station k and the MEC server.

The IoT devices can powered by the energy AP with the dedicated wireless power transmitter which can convert the received RF signal to a direct current (DC) signal, then store the energy to the battery. The harvested energy by IoT device is proportional to the received RF power from energy AP [17]. In each time slot, IoT devices can simultaneously perform energy harvesting when processing tasks by locally executing or offloading to the small base station and MEC server. About the loss of the energy harvested by IoT device from energy AP, we will discuss in Section V.A.

The arrival and consumption of renewable energy are time-varying; thus, we design the renewable energy arrival model by energy queuing, and the capacity of the obtained energy can be defined as the length of the queue. At the beginning of the time slot t, the length of the energy queue is denoted by $B_k(t)$. Then, the queue model can be expressed as

$$B_k(t+1) = max\left\{B_k(t) - E_k(t) + A_k^{(E)}(t), 0\right\} \tag{7}$$

According to the energy consumption model, it can be seen that in each time slot t, the static baseline power $P_{s,k}$ of the base station k is constant. For the wireless transmission power, when $B_k(t) > 0$, the small base station k is powered by renewable energy. When $B_k(t) \leq 0$, the renewable energy is insufficient. To ensure sustained operation of the small base station, the power grid is used for power supply. However, this increases the cost of energy consumption.

Regarding macro base station of the MEC, the static offset of power is stable, and the energy consumption of the macro base station changes little too, so the total cost of the MEC system is not affected by macro base station. We do not consider the consumed energy and computing delay while the MEC server transmits the computing results back to IoT terminals because the energy consumption of MEC is stable and the computing size of the backhaul is slight compared with transmitting to MEC.

C. BANDWIDTH ALLOCATION MODEL

The IoT devices in the service equipment layer should first communicate with small base station, then offload the task data to the MEC server, and finally perform computing offloading by the MEC server. We assume that the bandwidth of the small base station k is B_k in the time slot t. τ_n is the proportion of the bandwidth allocated to IoT device n. By Shannon's theorem, the data transmission rate between IoT device n and small base station k can be written as

$$r_{k,n}(t) = \tau_n B_k \log_2\left(1 + \frac{P_{k,n} h_{k,n}}{\sum_{i\in\mathcal{N}, i\neq n} p_{k,i} h_{k,i} + \sigma^2}\right) \tag{8}$$

where $P_{k,n}(t)$ is the transmission power of the IoT device n, $h_{k,n}(t)$ is the channel power gain from the n-th IoT device to the small base station k, and σ is the noise power at the small base station.

We assume that the bandwidth of the macro base station in the MEC layer is B_m, τ_k is the proportion of bandwidth allocated to the small base station k, so that the maximum allocated sub-bandwidth of the small base station k is B_m in the time slot t. The maximum achievable data transmission rate between the small base station k and the macro base station m is

$$r_{m,k}(t) = \tau_k B_m \log_2(1 + \zeta_k(t)) \tag{9}$$

D. COMPUTING MODEL

As the arrival of computing tasks is time-varying, we design a buffer queue system as the data buffer. Namely, the computing tasks of IoT devices are first pushed into the buffer queue, and then choose to pop the data from the buffer queue when there are tasks for local execution or offloading to small base station or the MEC server.

1) ARRIVAL OF COMPUTATION TASKS

We assume that the arrival of computing tasks at the IoT device n follows the Poisson distribution with $\lambda_n^{(D)}$. The arrival of computing tasks in time slot t can be processed in the next time slot $t+1$. As can be seen, there are $A_n^{(D)}(t)$ (bits) of computation tasks that arrive at the n-th IoT device in time slot t. We assume $A_n^{(D)}(t)$ in different time slots are i.i.d., which implies that $\mathbb{E}\left[A_n^{(D)}(t)\right] = \lambda_n^{(D)}$. Then, the total size of tasks $A_k^{(D)}(t)$ that should be processed at the buffer queue of small base station k in time slot t can be written as

$$A_k^{(D)}(t) = \sum_{n=0}^{N} A_n^{(D)}(t) \tag{10}$$

2) LOCAL EXECUTION

When there is a computing task request for an IoT device, and the computation capacity of the device is sufficient, the task can be processed by local computing. For the IoT devices n, let f_n be the clock frequency of the CPU and L_n the number of clock cycles required for the execution of tasks that can be availed by off-line measurements [36]. In time slot t, the number of computing bits for local execution is

$$D_{l,n}(t) = \rho_n \cdot \frac{\left[P_{l,n}(t)\right]^{1/3} f_n}{L_n} \tag{11}$$

where ρ_n is the weight coefficient, which depends on the computation capability of the IoT device. $P_{l,n}$ is the allocated local execution power [37], [38].

3) COMPUTATION OFFLOADING

As the coverage of small base stations in the service equipment layer does not overlap, each computing task for offloading is assigned to only one small base station. When a computing task is selected for offloading to the small base station k, the size of computing tasks offloaded from the IoT device n to the small base station k in time slot t is

$$D_{k,n}(t) = \begin{cases} r_{k,n}(t) * t, & \tau_n > 0 \\ 0, & \tau_n = 0 \end{cases} \tag{12}$$

where τ_n is the proportion of bandwidth allocated to IoT device n. When the bandwidth of the small base station is insufficient, the IoT devices cannot offload computing tasks. Thus, the total size of computing tasks offloaded from all IoT device to the small base station in time slot t is

$$D_k(t) = \sum_{n=1}^{N} D_{k,n}(t) \tag{13}$$

The MEC server has usually sufficient computing power and can rapidly process offloading tasks and even handle multi-tasking. Therefore, we neglect the processing and transmission delay (sending the computing results back to IoT devices) of the MEC server because the computing size of the backhaul is too small compared with that of task offloading to the MEC. That is, the total size of task offload to the MEC server in time slot t is

$$D_{m,k}(t) = \begin{cases} r_{m,k}(t) * t, & \tau_k > 0 \\ 0, & \tau_k = 0 \end{cases} \tag{14}$$

where τ_k is the proportion of bandwidth allocated to the small base station k. When the bandwidth of the MEC server is insufficient, the small base station cannot offload the task. Thus, the total size of computing tasks offloaded from all small stations to the MEC server in time slot t is

$$D_m(t) = \sum_{k=1}^{K} \left(D_{m,k}(t)\right) \tag{15}$$

According to Little's law, the average size of computing tasks in the queue system is equal to the average arrival rate multiplied by the average time that the task stays in the queue system. We consider utilizing the average length of the task buffer queue to represent the average delay of computing tasks. In time slot t, the total computing tasks of the MEC system are

$$D(t) = \sum_{k=1}^{K}\sum_{n=1}^{N} D_{l,n}(t) + \sum_{k=1}^{K} \xi_k D_k(t) + (1-\xi_k) D_m(t) \tag{16}$$

The arrival rates of the computing tasks for IoT device n follow the Poisson distribution with $\lambda_n^{(D)}$. If $L(t)$ denotes the length of the computing task buffer queue in time slot t, then the length of the computing task buffer queue in time slot $t+1$ is

$$L(t+1) = \begin{cases} L(t) - D(t) + \sum_{k=1}^{K} A_k^{(D)}(t), & 0 \le L(t) \le S \\ S, & L(t) > S \\ 0, & L(t) < 0 \end{cases} \tag{17}$$

where S is the upper bound of the buffer queue and ξ_k is the ratio of data offloading to the small base station k or MEC.

IV. PROBLEM STATEMENT AND RL

A. PROBLEM STATEMENT

The objective of the proposed MEC model is to minimize the total cost of the consumed energy, bandwidth utilization, and the length of the buffer queue under the constraint conditions. The optimization scheme can be summarized as follows: In time slot t, there are the task requests of the IoT node n, and these tasks are pushed into the buffer queue $L(t)$. Depending on the length of the energy queue, the tasks are either executed locally or offloaded to the small base station or the MEC server. For the energy consumption of the proposed MEC model, we focus on the consumed power for local

execution as well as the transmitting power for offloading to the small base station or MEC server. We define the energy consumption as the sum of the allocation power for local execution and the transmission power of IoT devices and the small base station k. The cost of the consumed energy, allocated bandwidth, and the length of the buffer queue under the constraint conditions is

$$\min \frac{1}{T}\sum_{t=0}^{T}\left[\omega_1\sum_{k=0}^{K}\left(E_k(t)+\tau_k B_m(t)+\sum_{n=1}^{N}\tau_n B_k(t)\right)+(1-\omega_1)L(t)\right] \tag{18a}$$

$$\text{s.t. } P_{l,n}(t)\le P_{max}^{(l)},\quad P_{k,n}(t)\le P_{max}^{(k)},\ P_{m,k}(t)\le P_{max}^{(m)} \tag{18b}$$

$$0\le\tau_k\le 1 \tag{18c}$$

$$0\le\tau_n\le 1 \tag{18d}$$

$$0\le L(t)\le S \tag{18e}$$

$$0\le\xi_k\le 1 \tag{18f}$$

where, ω_1 is an adjustment coefficient satisfying $0\le\omega_1\le 1$ and $E_k(t)$ can be obtained by (5). (18b) denotes the total energy consumption constraint for the IoT devices and the small base stations. (18c) is the limiting condition for the ratio of bandwidth allocation on offloading to the MEC server. (18d) is the limiting condition for the ratio of bandwidth allocation on offloading to the small base station k. (18e) ensures that the buffer overflow may not occur when the computing tasks arrive. (18f) denotes the ratio of computing offloading to the small base station or the MEC server.

B. DEEP REINFORCEMENT LEARNING OPTIMIZATION SCHEME

In the proposed MEC model, the IoT devices choose either local execution or offloading to the small base station or the MEC server. The arrival of computing tasks and renewable energy are stochastic and the wireless channel is time-varying. Accordingly, we do not select traditional machine-learning methods for computing offloading with labelled historical data for supervised learning. For learning the optimal computing offloading policy, we use DRL method to learn the optimization policy under stochastic tasks, renewable energy, and the wireless channel state. We use the DDPG algorithm for adaptively learning the optimal policy for local execution and task offloading based on continuous action space. DQN [39] is the baseline algorithm of deep reinforcement learning. It is widely used in various optimization areas. DDQN [40] and Dueling DQN [41] are improvements of DQN. However, the action spaces of DQN, DDQN, and Dueling DQN are discrete. For the proposed MEC model, the action space is continuous and which is require to discretize when DQN, DDQN or Dueling DQN are selected. If the action space is excessively large, the curse of dimensionality will occur. But DDPG can effectively handle continuous action space [42] and is an extension of the actor-critic algorithm. The actor network updates the parameters of the neural network by deterministic policy gradient to determine the optimal action in the current state. The critic network uses the time difference error to evaluate the policy of the actor network. Using the Q-function, the critic network updates the parameters of the neural network. DDPG is different from traditional stochastic policy gradient by choosing actions based on action distribution. DDPG ensures that a smaller amount of data is sampled, and the efficiency of the algorithm can be improved. To obtain the tradeoff between exploration and exploitation, DDPG uses the structure of DQN in the critic network and off-policy for sampling.

In time slot t, the actor selects an action a_t by the deterministic policy μ:

$$a_t=\mu\left(s_t|\theta^{\mu}\right)+\mathcal{N}_t \tag{19}$$

where θ^{μ} is the parameter of the actor's target network and $\mathcal{N}_t$ is stochastic noise for encouraging exploration.

Based on the deterministic policy gradient, the parameters of the actor neural network can be updated as follows:

$$\nabla_{\theta^{\mu}}J\left(\mu_\theta\right)=E_{S_t\sim\rho^{\mu}}\left[\nabla_{\theta^{\mu}}\mu_\theta\left(s_t\right)\nabla_{a_t}Q^{\mu}\left(s_t,a_t\right)|_{a_t=\mu_s(s_t)}\right] \tag{20}$$

We note that the deterministic policy gradient does not consider the action a_t and only samples s_t. DDPG updates the parameter of the actor network $\nabla_\theta\mu_\theta\left(s_t\right)$ and the parameter of the critic network $\nabla_{a_t}Q^{\mu}\left(s_t,a_t\right)$.

Then, the parameters of the critic network can be updated by

$$L=\mathbb{E}_{(s_t,a_t,r_t,s_{t+1})\sim M}\left(\left(y_t-Q\left(s_t,a_t|\theta^{Q}\right)\right)^2\right) \tag{21}$$

$$y_t=r_t+\gamma Q'\left(s_{t+1},\mu'\left(s_{t+1}|\theta^{\mu'}\right)|\theta^{Q'}\right) \tag{22}$$

To obtain the optimal policy for task offloading, energy allocation, and bandwidth allocation, we define state, action and reward of RL.

1) STATE SPACE

In time slot t, the agent observes the network environment, and either local execution or offloading to the small base station or MEC server will be selected for the computing tasks from the IoT devices. We assume that the channel vectors can be obtained by (1), (2), the size of the data to be offloaded changes with the channel vector and the allocated bandwidth in the next time slot. The agent can learn and predict the channel state. The bandwidth provided by small or macro base stations in the MEC server varies in each time slot. For optimizing the policy of RL, we denote $\mathcal{S}$ as the state space, and the state s_t can be defined by the following parameters:

$$\begin{aligned}s_t=\{&B_1(t),B_2(t),\ldots,B_k(t),\ldots,B_K(t),\\&\beta L(t),\\&H_{m,1}(t),H_{m,2}(t),\ldots,H_{m,k}(t),\ldots,H_{m,K}(t),\\&\zeta_1(t),\zeta_2(t),\ldots,\zeta_k(t),\ldots,\zeta_K(t),\\&\xi_1(t),\xi_2(t),\ldots,\xi_k(t)\ldots,\xi_K(t)\}\end{aligned} \tag{23}$$

where

$B_k(t)$ is the length of the energy queue k.

$L(t)$ is the length of the buffer queue for computing tasks in the MEC system.

$H_{m,k}(t)$ is the channel vector between the small base station k and the MEC server for uplink transmission.

$\zeta_k(t)$ is the SINR of the small base station k.

$\xi_k(t)$ is the offloading ratio to the small base station k or MEC server.

2) ACTION SPACE

In time slot t, the agent selects an action a_t from the action space according to the value of state s_t. If the policy of offloading to the small base station or the MEC server is chosen, the power and bandwidth are allocated to IoT devices. If local execution is chosen, the local power is consumed to execute the task. In the proposed MEC model, the agent will select the values of allocated power, transmitted power, and allocated bandwidth for the IoT nodes in each time slot. The action space is denoted as $\mathcal{A}$ and which is continuous. The action a_t in time slot t can be defined as

$$\begin{aligned} a_t = \{&E_{l,1}(t), E_{l,2}(t), \ldots, E_{l,n}(t), \ldots, E_{l,N}(t), \\ &E_{k,1}(t), E_{k,2}(t), \ldots E_{k,n}(t), \ldots, E_{k,N}(t), \\ &E_{m,1}(t), E_{m,2}(t), \ldots E_{m,k}(t), \ldots, E_{m,K}(t), \\ &\tau_1(t), \tau_2(t), \ldots, \tau_n(t), \ldots, \tau_N(t), \\ &\tau_1(t), \tau_2(t), \ldots, \tau_k(t), \ldots, \tau_K(t)\} \end{aligned} \tag{24}$$

where

$E_{l,n}(t)$ is the energy consumption of the IoT device n for local execution which can be obtain by $P_{l,n}(t) * t$.

$E_{k,n}(t)$ is the energy consumption of the IoT device n for offloading to the small base station k which can be obtain by $P_{k,n}(t) * t$.

$E_{m,k}(t)$ is the energy consumption of the small base station k for offloading to the MEC server m which can be obtain by $P_{m,k}(t) * t$.

$\tau_n(t)$ is the ratio of bandwidth allocated to the IoT device n for offloading to the small base station k.

$\tau_k(t)$ is the ratio of bandwidth allocated to the small base station k for offloading to the MEC server.

It should be noted that we do not define the consumed power of MEC for processing the offloaded tasks from a small base station, as explained in detail in Section III.B.

3) REWARD FUNCTION

The agent interacts with the environment and selects the action a_t according to the input state s_t, and the immediate reward can be obtained then added to the cumulative rewards of the previous state s_{t-1}, the current cumulative reward can be obtained. The goal of RL is to maximize the expected cumulative rewards. In general, the reward function is related to the cost function. The agent of RL wants to find the optimal policy for minimizing the cost of the proposed system, which consists of the energy consumption, the allocated bandwidth, and the length of the task buffer queue. According to the energy model, the energy consumed by the IoT devices and the small base stations includes the transmission power for local execution and offloading to small base stations or the MEC server. That is, the reward function includes the payment for renting bandwidth, the cost for energy consumption, and the cost for the task buffer queue. We assume that the payment for renting bandwidth is proportional to the bandwidth. In the time slot t, we can define the reward r_t under the given conditions as follows:

$$\begin{aligned} r_t = -\mathbb{E}\Bigg[\Bigg(\omega_1 \Bigg[\alpha_1 \sum_{k=1}^{K} [E_k(t)] + \alpha_2 \sum_{k=1}^{K} [\tau_k B_m(t)]\Bigg] \\ \alpha_3 \sum_{k=1}^{K} \sum_{n=1}^{N} [\tau_n B_n(t)] + (1-\omega_1)\,\alpha_4 L(t)\Bigg) - P(t)\Bigg] \end{aligned} \tag{25}$$

where ω_1 is a coefficient satisfying $\omega_1 \leq 1$ and is used to adjust the penalty ratio for the energy consumption, allocated bandwidth of IoT devices and small base station, and task buffer delay of the MEC system. $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ are the weight parameters of the consumed energy $E_k(t)$, the allocated bandwidth $B_m(t)$, $B_n(t)$, and the length of the task buffer queue $L(t)$, respectively, and $P(t)$ is a negative penalty factor for buffer overflows. To improve the convergence of proposed DRL algorithm, the value of r_t is defined as negative. We define the cost of energy consumption, bandwidth, and buffer queue delay as the negative of rewards. Minimizing the cost amounts to maximizing the expected cumulative rewards, that is,

$$R = \max\left[\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} r_t\right] \tag{27}$$

The details of the proposed algorithm are shown in Algorithm 1.

V. NUMERICAL RESULTS

In this section, we will evaluate the performance of the proposed joint optimization scheme for data transmission delay, energy consumption, and bandwidth allocation based on JODRBRL through computer simulations.

A. SIMULATION SETUP

The simulations are run on a PC with an Intel Core i9-9900K CPU at 3.6 GHz. The graphics card is the NVIDIA GeForce RTX2080Ti with 11 GB memory. We use Tensorflow 1.10.0 with Python 3.6 on Ubuntu 16.04 LTS in the simulations to implement JODRBRL and compare JODRBRL to three benchmark algorithms. In the simulations, we assume N IoT devices are located at an equal distance of 20 m from the small base station k and 80 m from the MEC server. There are 4 IoT devices and 2 small base stations. Each small base station covers 2 IoT devices. Moreover, $t = 2$ ms, $\sigma^2 = 10^{-9}$, $\rho_c = 0.95$, $f_n = 1.2$ GHz, $L_n = 1200$, $S = 100$, $P = 50$ and $\eta = 1$. The maximum allocated power for IoT devices $P_{max}^{(l)}$ is 0.5 W. The maximum transmission power of an IoT device

Algorithm 1 Joint Optimization Method Based on the Proposed JODRBRL

1: Initialize the parameters of the actor's and critic's main networks: θ^{μ}, θ^{Q};
2: Soft copy all the parameters of the main network of the actor and the critic to the target network of the actor and the critic:$\theta^{\mu'} \leftarrow \theta^{\mu}$, $\theta^{Q'} \leftarrow \theta^{Q}$
3: Initialize the experience replay memory M;
4: **for** episode $= [1, 2, \cdots, E_{max}]$ **do**
5: Reset the simulation environment of the proposed MEC model;
6: Reset the state s_0: including the length of the renewable energy queue, the length of the computing task buffer queue, the channel vector, the SINR, and the offloading ratio to a small base station or MEC server for all n IoT devices; reset $r_0 = 0$;
7: **for** $[t = 1, 2, \cdots, T]$ **do**
8: Reset the OU noise vector $\mathcal{N}_t$;
9: According to the policy μ, select a_t in line with (19), including the average consumed power for local execution and offloading to a small base station and MEC server, the allocated bandwidth, and the OU noise vector $\mathcal{N}_t$;
10: Choose a_t in the simulation environment and obtain the reward r_t and the next state s_{t+1};
11: Store the transition tuple $< s_t, a_t, r_t, s_{t+1} >$ in experience replay memory M as the training dataset for the main network of the actor and the critic;
12: Randomly sample N transition tuples from experience replay memory M as mini-batch data for training the main network of the actor and the critic;
13: According to (21) and (22), calculate the gradient $\nabla_{\theta^{\varrho}} L$ of the critic's main network;
14: Update the parameters θ^{Q} of the critic's main network using the Adam optimizer;
15: According to (20), calculate the policy gradient $\nabla_{\theta^{\mu}} J(\mu_{\theta})$ of the actor's main network;
16: Update the parameters θ^{μ} of the actor's the main network using the Adam optimizer;
17: Soft update the parameters $\theta^{Q'}$, $\theta^{\mu'}$ of the actor's target network and the critic's target network;
18:

$$\begin{cases} \theta^{Q'} \leftarrow \tau\theta^{Q} + (1-\tau)\theta^{Q'} \\ \theta^{\mu'} \leftarrow \tau\theta^{\mu} + (1-\tau)\theta^{\mu'} \end{cases} \tag{26}$$

where $\tau = 0.001$
19: **end for**
20: **end for**

to small base station $P_{max}^{(k)}$ is 0.5 W. The MEC server $P_{max}^{(m)}$ is 0.5 W. The static power of each small base station $P_{s,k}$ is 0.5 W. The maximum bandwidth of the MEC server and small base station is respectively 6 MHz and 3 MHz.

For JODRBRL, Dueling DQN, and DDQN, we construct fully connected neural networks with an input layer, two hidden layers, and an output layer. We take advantage of two advanced techniques for the three algorithms, including fixing the parameters of the target network in a certain period of time and experience replay memory which can obviously improve the convergence of the algorithms. Regarding the main hyperparameters of the neural networks, the number of neurons in two hidden layers is set to 200 and 100. The learning rate of the actor and critic is set to 10^{-4} and 10^{-3}, respectively, and the learning rate of Dueling DQN and DDQN is set to 10^{-3}. The size of the experience replay memory is 200000. The action bound and the state bound are set to 5. The value of the penalty coefficient α_1 is related to renewable energy. According to section III.B, the energy model is described in detail. Renewable energy supply can greatly reduce energy consumption, but we should consider the loss of the energy harvested by IoT devices from energy AP. So, if the renewable energy supply for IoT devices and small base stations is sufficient, we do not set the penalty coefficient for energy consumption α_1 to 0. Considering the loss of energy harverst and transmission, α_1 is set to 0.5. When there are no sufficient renewable energy supply, α_1 is set to 1. α_2, α_3, α_4 of from the reward function are set to 0.5,0.5, 0.05, respectively.

B. PERFORMANCE COMPARISON

We compare the performance of JODRBRL with that of three benchmark algorithms: Dueling-DQN, DDQN, and greedy policy.

However, the action space of Dueling DQN and DDQN is discrete. Thus, we should quantize the continuous-valued actions in the simulation environment. The discrete value is an approximation to the real continuous values. We implemented the joint optimization algorithm based on Dueling DQN, DDQN, and uniformly quantized a_t from the action space into 10 levels.

1) DUELING DQN

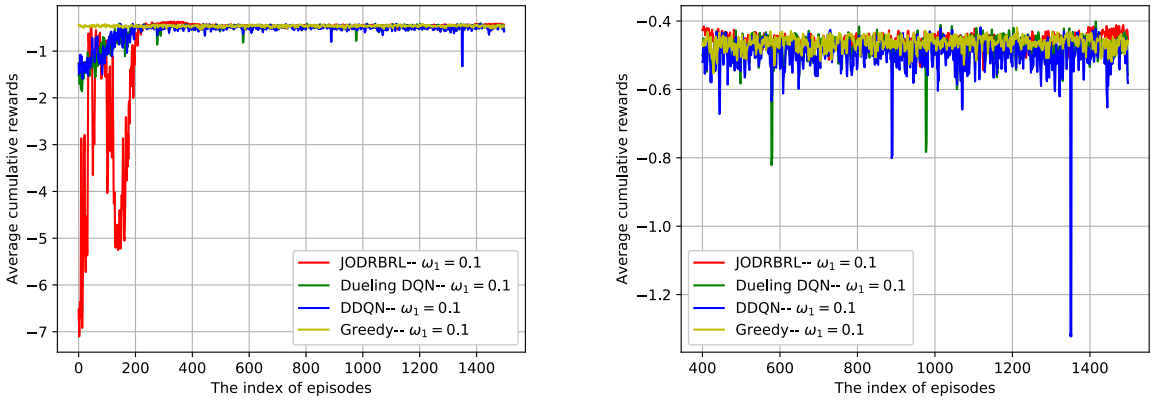
Dueling DQN divides the state-action function $Q(s_t, a_t)$ into two parts: the state value function $V(s_t, a_t)$ and the advantage function $A(s_t, a_t)$. $V(s_t, a_t)$ is independent of the action a_t, and $A(s_t, a_t)$ represents the advantage of selecting action a_t in state s_t.

2) DDQN

DDQN realizes action selection and evaluation by different value functions to resolve the over-estimation of DQN.

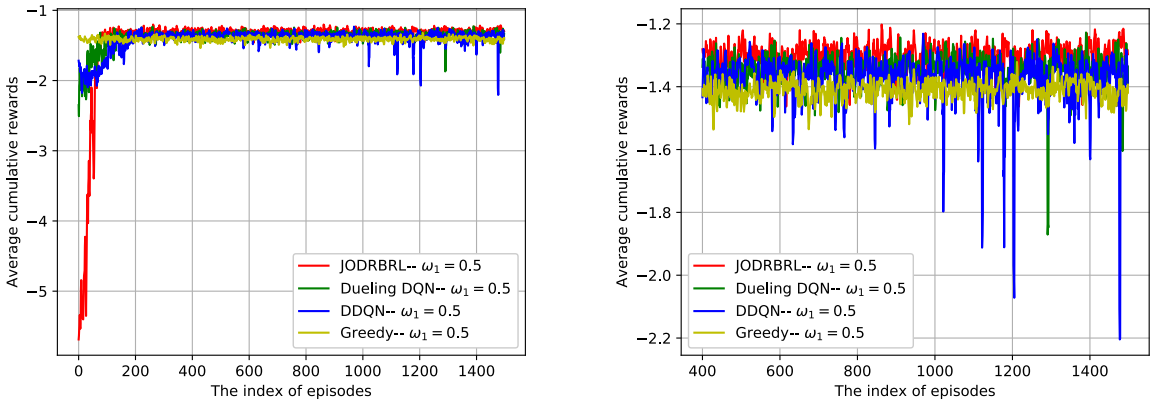
3) GREEDY POLICY

In time slot t, the greedy policy chooses to execute computing tasks at maximum power and bandwidth to ensure that the length of the data buffer queue is minimized regardless of whether local execution or offloading is selected.



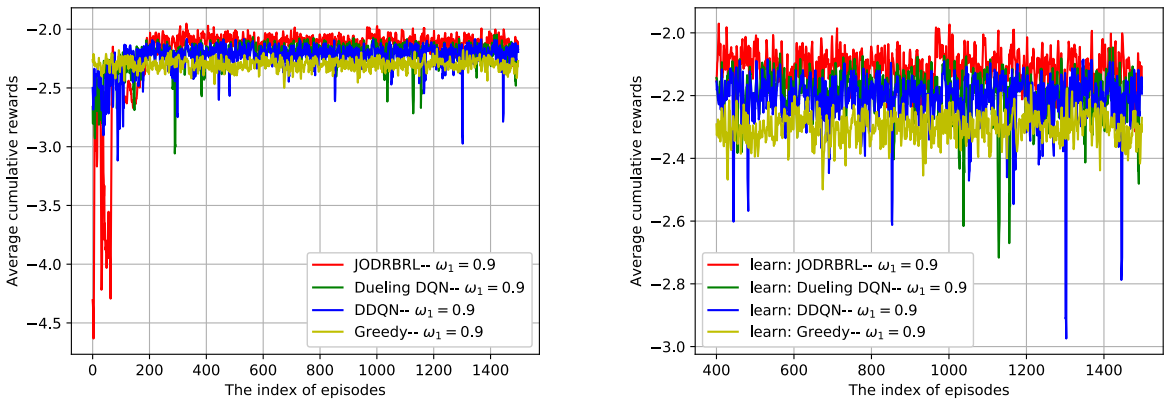
(a) Result of all 1500 episodes (b) Result after 400 episodes

FIGURE 2. Average cumulative reward achieved by different algorithms when $\omega_1 = 0.1$.



(a) Result of all 1500 episodes (b) Result after 400 episodes

FIGURE 3. Average cumulative reward achieved by different algorithms when $\omega_1 = 0.5$.



(a) Result of all 1500 episodes (b) Result after 400 episodes

FIGURE 4. Average cumulative rewards achieved by different algorithms when $\omega_1 = 0.9$.

C. SIMULATION RESULTS

Figure 2, Figure 3, and Figure 4 show the average cumulative rewards curves for JODRBRL and the other algorithms obtained by different values of the adjust coefficient. The adjust coefficient ω_1 is set to 0.1, 0.5, and 0.9 respectively in Fig. 2, Fig. 3, and Fig. 4. The arrival rate $\lambda_n^{(D)}$ of computing tasks for IoT device n is set to 0.75, and the arrival rate $\lambda_k^{(E)}$ of renewable energy for the small base station k is set

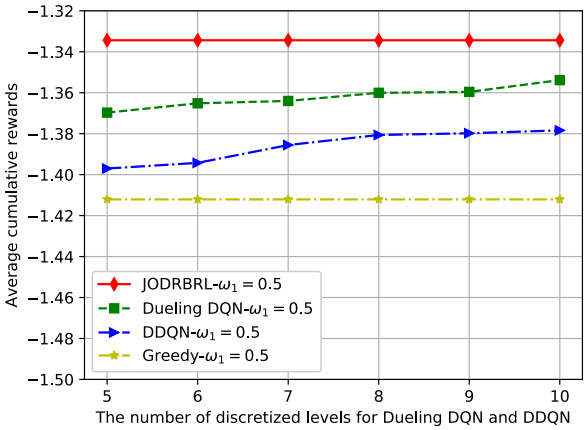


FIGURE 5. Comparison on the number of discretization levels for Dueling DQN and DDQN.

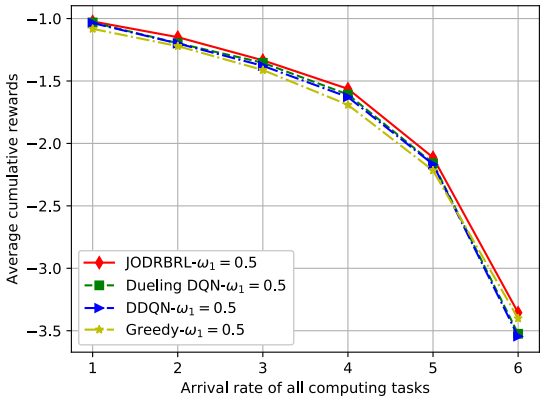
to 1. As shown in Fig. 2, when $\omega_1 = 0.1$, the cost of the average consumed energy and allocated bandwidth is smaller, whereas the cost of the task buffer delay is greater. Although greedy policy handles computing data at the maximum transmission power and bandwidth, the average cumulative rewards are greater than that of the other three algorithms. However, the average cumulative rewards of JODRBRL is nearly equal to that of the greedy policy. As shown in Fig. 3, when $\omega_1 = 0.5$, the cost of the average consumed energy and allocated bandwidth equals to the cost of the task buffer delay. The performance of JODRBRL is better than that of Dueling DQN, DDQN, and greedy policy owing to the continuous action space of JODRBRL. As shown in Fig. 4, when $\omega_1 = 0.9$, the cost of the average consumed energy and allocated bandwidth is far greater, and greedy policy has the worst performance. Although the performance of Dueling DQN and DDQN is close to that of JODRBRL with different adjustment coefficient ω_1, the discretization of the action space for Dueling DQN and DDQN affect performance in the simulations.

We compare the performance of JODRBRL, Dueling DQN, DDQN, and greedy policy in Fig. 5, where the action space of Dueling DQN and DDQN is discrete. We set $\omega_1 = 0.5$ and uniformly quantize the average consumed energy and allocated bandwidth into 5 to 10 levels in this simulation because the action space includes the average consumed power for the allocated power for local execution, the average consumed power (transmission power of IoT devices to small base stations and the MEC server), and the allocated bandwidth for offloading to small base stations and the MEC server. To ensure the performance of Dueling DQN and DDQN, the maximum number of discretization levels is set to 10. As shown in Fig. 5, the average cumulative rewards of Dueling DQN and DDQN increases with the number of discretized levels, but which are still less than that of JODRBRL. However, as the action space of JODRBRL is continuous, the actions need not be discretized. Greedy policy processes computing tasks at maximum power and bandwidth and the rewards does not change with the number of discretized levels. That is, the average cumulative rewards of JODRBRL and greedy policy is fixed.

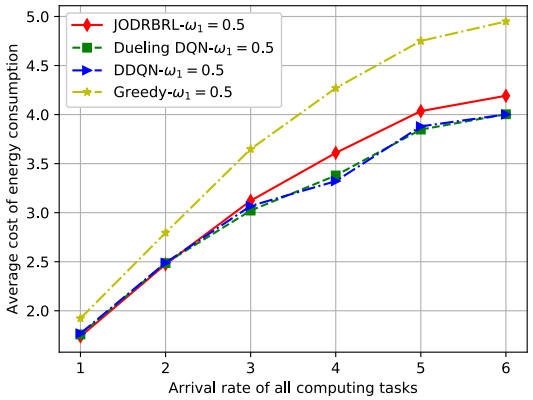
Figure 6 shows the performance of the four algorithms under different task arrival rates. We set the parameter ω_1 to 0.5, the renewable energy arrival rate $\lambda_k^{(E)}$ for the small base station k is set to 1, and the total renewable energy rate is 2. We set the range of the total task arrival rate to $[1, 6]$. In this simulation, we assumed that the task arrival rate for each IoT device is the same. Fig. 6(a) shows the comparison of the average cumulative rewards for the four algorithms under different task arrival rates. Lower task arrival rate implies greater average cumulative rewards for all IoT devices. This is because the size of task data that should be processed increases with the arrival rate. The length of data buffer queue is greater, and more energy is consumed. JODRBRL outperforms Dueling DQN, DDQN, and greedy policy in terms of the average cumulative rewards. When the total task arrival rate is 5 or 6, the average cumulative rewards of Dueling DQN and DDQN is less than that of greedy policy. Power and bandwidth discretization in the action space affects the performance of Dueling DQN and DDQN. As shown in Fig. 6(b), in terms of the average cost of consumed energy, greedy policy chooses to process computing tasks at the maximum energy and bandwidth for local execution and offload to small base stations and MEC server, and the average cost of energy consumption of all IoT devices is greater than for other algorithms. Although the cost of energy consumption for JODRBRL is greater than that of Dueling DQN and DDQN when the total task arrival rate is 3, 4, 5, and 6, JODRBRL uses joint optimization to obtain the maximum cumulative rewards. Fig. 6(c) shows the comparison of the average length of the task queue for different algorithms. The average length of the task queue increases with the arrival rate of all IoT devices. As greedy policy executes computing tasks at the maximum energy and bandwidth, the queue length is less than that of the other three algorithms. However, in terms of average cumulative rewards, the performance of greedy policy is worse than that of the other three algorithms. Dueling DQN and DDQN can decrease the energy consumption as much as possible but do not consider the length of the buffer queue for the optimal policy.

In order to verify the effectiveness of the renewable energy supply, when there are no renewable energy, we show the effectiveness of JODRBRL and other three algorithms in Fig. 7. Further more, we compare JODRBRL under renewable energy arrival rates $\lambda_k^{(E)} = 1$ with DRL without the renewable energy supply which is shown in Fig. 8.

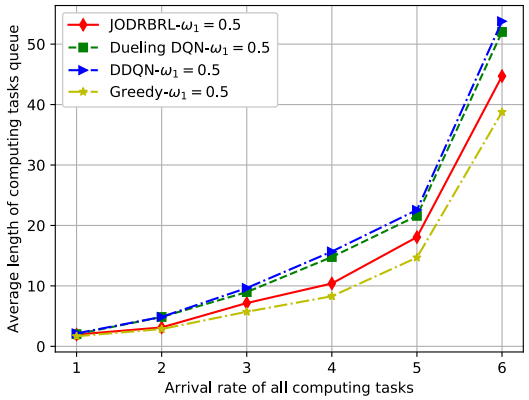
As shown in Fig. 7, we set the parameter $\omega_1 = 0.5$, the task arrival rate of IoT device n $\lambda_n^{(D)} = 0.75$. So the arrival rate of all computing tasks is 3. There are no renewable energy supply for MEC model and all IoT devices, small station station and MEC server is powered by the power grid. Therefore, α_1 is set to 1. the penalty for energy consumption is great. Although there is no renewable energy supply, the performance of JODRBRL is still better than the other



(a) Average cumulative rewards



(b) Average cost of energy consumption



(c) Average data queue length

FIGURE 6. Performance of four algorithms under different task arrival rates.

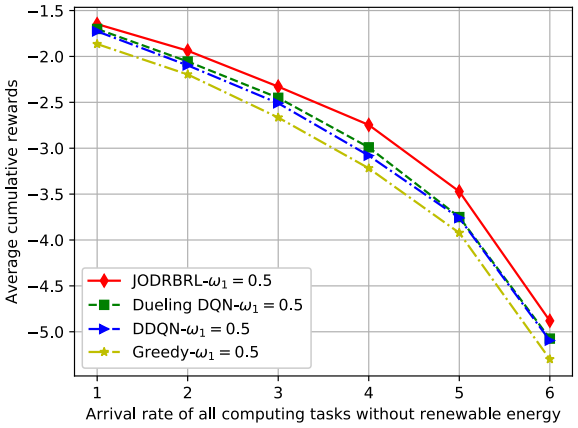
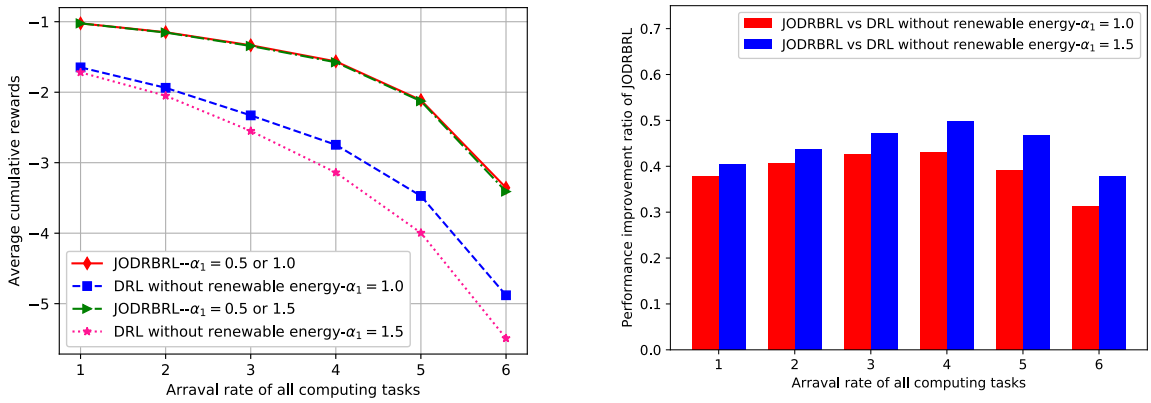


FIGURE 7. Comparison on the performance of four algorithms without renewable energy.

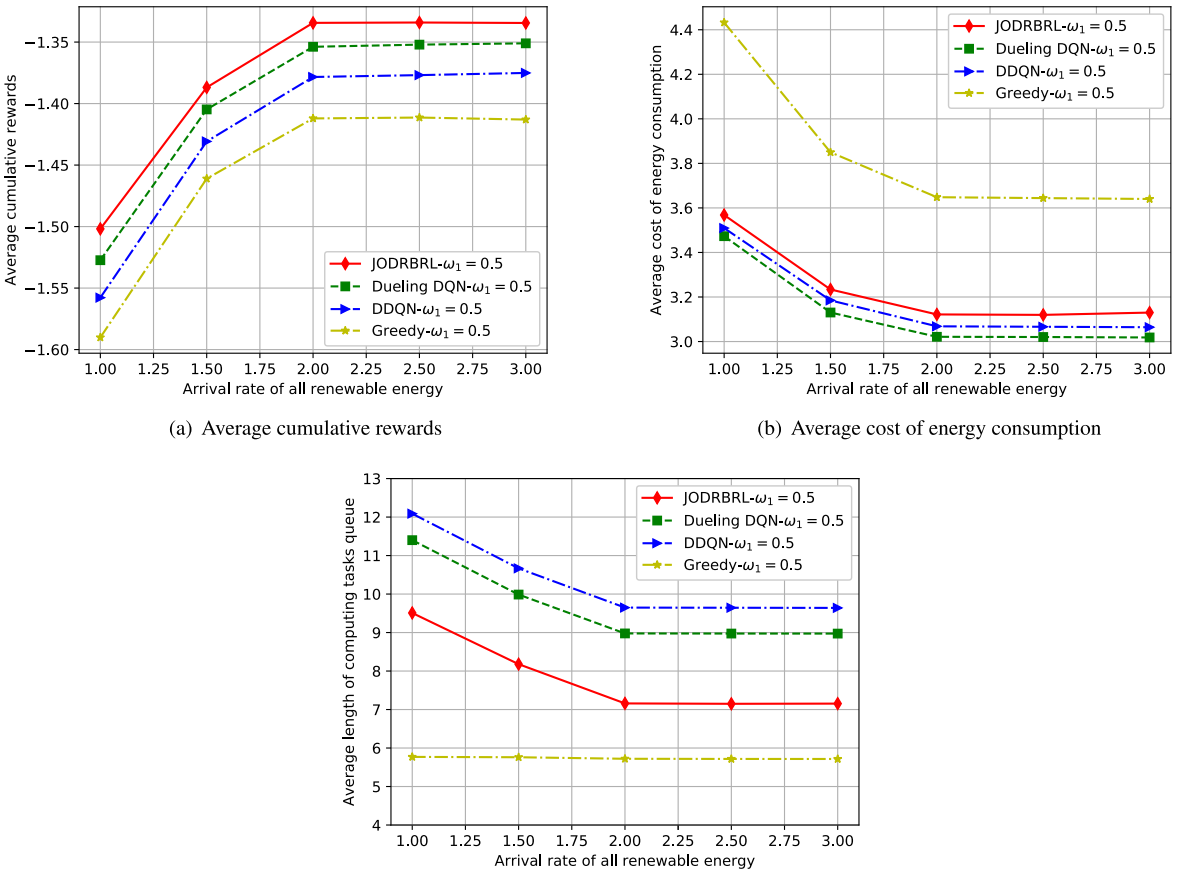
three algorithms in terms of average cumulative rewards(the cost of MEC system). Because the cost of energy consumption is greater than that of JODRBRL, Dueling DQN and DDQN, the performance of the greedy policy is the worst regardless of what the value of the task arrival rate is.

Figure 8 demonstrates the performance of JODRBRL compare to DRL without the renewable energy supply. We set the parameter $\omega_1 = 0.5$, the total task arrival rate is 3. The only difference between DRL and JODRBRL is that there are no renewable energy supply for DRL in the proposed MEC model, and the rest processing of DRL is the same as JODRBRL. In order to compare DRL without the renewable energy supply with JODRBRL, we set α_1 to 1, 1.5 respectively for DRL. For JODRBRL, the total arrival rate of renewable energy is set to 2 and α_1 is set to 0.5 when there is sufficient renewable energy supply. As shown in Fig 8(a), although α_1 is 1 or 1.5, in terms of average cumulative rewards, JODRBRL outperforms DRL algorithm without the renewable energy supply. When $\alpha_1 = 1.5$, the cost of energy consumption is great. The performance of DRL is worse than that of $\alpha_1 = 1$. When the arrival rate of all computing tasks is 6, due to the exploration the average cumulative rewards of JODRBRL with $\alpha_1 = 1.5$ is slight less than that of $\alpha_1 = 1$. As shown in Fig 8(b), We plot the performance improvement ratio between JODRBRL and DRL without renewable energy. When the total computing tasks arrival ratio is set to 1, 2, 3 or 4, the performance improvement ratio gradually become greater when tasks arrival ratio increases. This is because the processing capacity of local proccessor and MEC server can meet requirement for the less tasks arrival rate. When the total computing tasks arrival



(a) Average cumulative rewards (b) Performance increase ratio

FIGURE 8. **Comparison on JODRBRL and DRL without renewable energy under different α_1.**



(a) Average cumulative rewards (b) Average cost of energy consumption

(c) Average data queue length

FIGURE 9. **Performance of four algorithms under different renewable energy arrival rates.**

rate is 5 or 6, the tasks in buffer queue need to be executed locally and offloaded to MEC server with great transmitting power and bandwidth. Therefore, the performance improvement ratio slightly decreases. In short, the cost of renewable energy supply is less than the power supply of the grid, so the performance of the JODRBRL algorithm is significantly better than DRL without the renewable energy supply.

Figure 9 shows the performance of the four algorithms under different renewable energy arrival rates. We set the parameter ω_1 to 0.5, the task arrival rate $\lambda_n^{(D)}$ for each IoT device is set to 0.75, and the total task arrival rate is 3. We set the range of the total renewable energy rate to [1, 3]. In this simulation, we assumed that the renewable energy arrival rate for all small base stations is the same. In the simulation setup, we analyzed that the value of the adjustment coefficient α_1 is

related to renewable energy. If the renewable energy supply to IoT devices and small base stations is sufficient, α_1 is set to 0.5; otherwise, it is set to 1. As shown in Fig. 9(a), when the total arrival rate of renewable energy is 2, 2.5, or 3, the renewable energy is sufficient. The average cumulative rewards, the cost of average energy consumption and the average length of the task queue do not significantly change and stabilize near a certain value. In terms of the average cumulative rewards under different renewable energy arrival rates regardless of whether the renewable energy is sufficient, the four algorithms are ranked as follows: JODRBRL, Dueling DQN, DDQN, and greedy policy. Fig. 9(b) shows the comparison of the average cost of energy consumption for the four algorithms. Although the cost of energy consumption for JODRBRL is greater than that of Dueling DQN and DDQN, the difference is small, and the cost of energy consumption for these algorithms is significantly less than that of greedy policy. Fig. 9(c) shows the comparison of the average length of the task queue for different algorithms. When the renewable energy arrival rate is 1 or 1.5, the renewable energy cannot meet the power supply demand, and the cost of the energy consumption becomes large. At this time, the average length of the buffer queue becomes larger. When the renewable energy arrival rate is greater than 2, the average length of the task queue does not significantly vary with the renewable energy arrival rate.

VI. CONCLUSION

In this study, we design an MEC framework in the heterogeneous cellular networks that included the communication of IoT devices with small base stations and the MEC server. Considering that the wireless channel condition is time-varying, renewable energy and computing tasks arrive randomly, and the action space is continuous, we proposed an joint optimization scheme for data transmission delay, energy consumption and bandwidth allocation based on JODRBRL, which is a model-free DRL framework. Interacting with the dynamic simulation environment, JODRBRL adaptively learned the optimal policy to minimize the cost of the energy consumption, bandwidth allocation, and data transmission delay. Simulation results demonstrated that JODRBRL has stable learning capacity and performance for various parameter configurations. The computing tasks handled in our work are computation-intensive which are required to execute locally and offload to small base station or MEC in time. In future work, when dealing with computing tasks that are not intensive or in sparse networks, we consider to use the duty cycle of IoT devices to make full use of the computing power and residual energy for improving system utilization.

REFERENCES

[1] C. Zhang, H. Zhao, and S. Deng, "Synthetic structure of industrial plastics," *IEEE Access*, vol. 6, pp. 73520–73530, 2018.
[2] J. Du, F. R. Yu, X. Chu, J. Feng, and G. Lu, "Computation offloading and resource allocation in vehicular networks based on dual-side cost minimization," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1079–1092, Feb. 2019.
[3] J. Feng, Z. Liu, C. Wu, and Y. Ji, "AVE: Autonomous vehicular edge computing framework with ACO-based scheduling," *IEEE Trans. Veh. Technol.*, vol. 66, no. 12, pp. 10660–10675, Dec. 2017.
[4] L. T. Tan and R. Q. Hu, "Mobility-aware edge caching and computing in vehicle networks: A deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 10190–10203, Nov. 2018.
[5] K. Kumar, J. Liu, Y.-H. Lu, and B. Bhargava, "A survey of computation offloading for mobile systems," *Mobile Netw. Appl.*, vol. 18, no. 1, pp. 129–140, Feb. 2013.
[6] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, Aug. 2017.
[7] Y. Sun, L. Xu, Y. Tang, and W. Zhuang, "Traffic offloading for online video service in vehicular networks: A cooperative approach," *IEEE Trans. Veh. Technol.*, vol. 67, no. 8, pp. 7630–7642, Aug. 2018.
[8] G. Xiao, H. Zhang, H. Hassan, Y. Chen, Z. Huang, and N. Sun, "A cooperative offloading game on data recovery for reliable broadcast in vanet," *Concurrency Comput. Pract. Exper.*, vol. 29, no. 14, p. e3938, Sep. 2017.
[9] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, "Delayoptimal computation task scheduling for mobile-edge computing systems," in *Proc. IEEE Int. Conf. Symp. Inf. Theroy (ISIT)*, Barcelona, Spain, Jul. 2016, pp. 1451–1455.
[10] 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, Dec. 2016.
[11] M. Pasha and K. U. Rahman Khan, "Scalable and energy efficient task offloading schemes for vehicular cloud computing," *Int. J. Comput. Netw. Commun.*, vol. 10, no. 6, pp. 35–52, Jan. 2018.
[12] C. Wang, Y. He, F. R. Yu, Q. Chen, and L. Tang, "Integration of networking, caching, and computing in wireless systems: A survey, some research issues, and challenges," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 7–38, Oct. 2018.
[13] Z. Chen and X. Wang, "Decentralized computation offloading for multi-user mobile edge computing: A deep reinforcement learning approach," Dec. 2018, *arXiv:1812.07394*. [Online]. Available: https://arxiv.xilesou.top/pdf/1812.07394.pdf
[14] Y. Mao, J. Zhang, Z. Chen, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3590–3605, Dec. 2016.
[15] F. Sun, F. Hou, N. Cheng, M. Wang, H. Zhou, L. Gui, and X. Shen, "Cooperative task scheduling for computation offloading in vehicular cloud," *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 11049–11061, Aug. 2018.
[16] X. Liu, A. Liu, T. Wang, K. Ota, M. Dong, Y. Liu, and Z. Cai, "Adaptive data and verified message disjoint security routing for gathering big data in energy harvesting networks," *J. Parallel Distrib. Comput.*, vol. 135, pp. 140–155, Jan. 2020.
[17] S. Bi, C. K. Ho, and R. Zhang, "Wireless powered communication: Opportunities and challenges," *IEEE Commun. Mag.*, vol. 53, no. 4, pp. 117–125, Apr. 2015.
[18] F. Zhou, Y. Wu, H. Sun, and Z. Chu, "UAV-enabled mobile edge computing: Offloading optimization and trajectory design," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kansas City, MO, USA, May 2018, pp. 1–6.
[19] M. Min, L. Xiao, Y. Chen, P. Cheng, D. Wu, and W. Zhuang, "Learning-based computation offloading for IoT devices with energy harvesting," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1930–1941, Feb. 2019.
[20] X. Liu, M. Zhao, A. Liu, and K. K. L. Wong, "Adjusting forwarder nodes and duty cycle using packet aggregation routing for body sensor networks," *Inf. Fusion*, vol. 53, pp. 183–195, Jan. 2020.
[21] S. Bi and Y. Zhang, "Computation rate maximization for wireless powered mobile-edge computing with binary computation offloading," *IEEE Trans. Wireless Commun.*, vol. 17, no. 6, pp. 4177–4190, Jun. 2018.
[22] Y. Sun, X. Guo, J. Song, S. Zhou, Z. Jiang, X. Liu, and Z. Niu, "Adaptive learning-based task offloading for vehicular edge computing systems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3061–3074, Jan. 2019.
[23] X. Lyu, H. Tian, W. Ni, Y. Zhang, P. Zhang, and R. P. Liu, "Energy-efficient admission of delay-sensitive tasks for mobile edge computing," *IEEE Trans. Commun.*, vol. 66, no. 6, pp. 2603–2616, Jun. 2018.
[24] S. Zhang, N. Zhang, S. Zhou, J. Gong, Z. Niu, and X. Shen, "Energy-aware traffic offloading for green heterogeneous networks," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1116–1129, May 2016.
[25] Z. Zhou, P. Liu, Z. Chang, C. Xu, and Y. Zhang, "Energy-efficient workload offloading and power control in vehicular edge computing," in *Proc. IEEE Wireless Commun. Netw. Conf. Workshops (WCNCW)*, Barcelona, Spain, Apr. 2018, pp. 191–196.

[26] X. Lyu, H. Tian, P. Zhang, and C. Sengul, "Multiuser joint task offloading and resource optimization in proximate clouds," *IEEE Trans. Veh. Technol.*, vol. 66, no. 4, pp. 3435–3447, Apr. 2016.
[27] J. Xu, L. Chen, and S. Ren, "Online learning for offloading and autoscaling in energy harvesting mobile edge computing," *IEEE Trans. Cogn. Netw.*, vol. 3, no. 3, pp. 361–373, Sep. 2017.
[28] Y. Wei, F. R. Yu, M. Song, and Z. Han, "Joint optimization of caching, computing, and radio resources for fogenabled IoT using natural actor-critic deep reinforcement learning," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2061–2073, Oct. 2018.
[29] L. Quan, Z. Wang, and F. Ren, "A novel two-layered reinforcement learning for task offloading with tradeoff between physical machine utilization rate and delay," *Future Internet*, vol. 10, no. 7, p. 60, Jul. 2018.
[30] Z. Li, C. Wang, and C.-J. Jiang, "User association for load balancing in vehicular networks: An online reinforcement learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 8, pp. 2217–2228, Aug. 2017.
[31] J. Wang, J. Hu, G. Min, W. Zhan, Q. Ni, and N. Georgalas, "Computation offloading in multi-access edge computing using a deep sequential model based on reinforcement learning," *IEEE Commun. Mag.*, vol. 57, no. 5, pp. 64–69, May 2019.
[32] 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.
[33] H. Q. Ngo, E. G. Larsson, and T. L. Marzetta, "Energy and spectral efficiency of very large multiuser MIMO systems," *IEEE Trans. Commun.*, vol. 61, no. 4, pp. 1436–1449, Apr. 2013.
[34] H. A. Suraweera, T. A. Tsiftsis, G. K. Karagiannidis, and A. Nallanathan, "Effect of feedback delay on amplifyand-forward relay networks with beamforming," *IEEE Trans. Veh. Technol.*, vol. 60, no. 3, pp. 1265–1271, Feb. 2011.
[35] Y. Wei, F. R. Yu, and M. Song, "User scheduling and resource allocation in HetNets with hybrid energy supply: An actor-critic reinforcement learning approach," *IEEE Trans. Wireless Commun.*, vol. 17, no. 1, pp. 680–692, Nov. 2017.
[36] A. P. Miettinen and J. K. Nurminen, "Energy efficiency of mobile clients in cloud computing," in *Proc. 2nd USENIX Conf. Hot Topics Cloud Comput. (HotCloud)*, Boston, MA, USA, vol. 10, Jun. 2010, pp. 1–7.
[37] K. De Vogeleer, G. Memmi, P. Jouvelot, and F. Coelho, "The energy/frequency convexity rule: Modeling and experimental validation on mobile devices," in *Proc. Conf. Parallel Appl. Math. (ICPPAM)*. Warsaw, Poland: Springer, Sep. 2013, pp. 793–803.
[38] T. D. Burd and R. W. Brodersen, "Processor design for portable systems," *J. VLSI Signal Process. Syst.*, vol. 13, nos. 2–3, pp. 203–221, Aug. 1996.
[39] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, and G. Ostrovski, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.
[40] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proc. AAAI*, Phoenix, AZ, USA, Feb. 2016, pp. 2094–2100.
[41] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, "Dueling network architectures for deep reinforcement learning," in *Proc. ICML*, New York, NY, USA, Jun. 2016, pp. 1995–2003.
[42] S. Gu, T. Lillicrap, I. Sutskever, and S. Levine, "Continuous deep Q-learning with model-based acceleration," in *Proc. ICML*, New York, NY, USA, Jun. 2016, pp. 2829–2838.

HONGCHANG KE received the B.S. and M.S. degrees from the College of Computer Science and Technology, Jilin University, in 2004 and 2007, respectively, where he is currently pursuing the Ph.D. degree in computer application technology. He is interested in topics related to wireless networks and vehicular networks, especially for offloading and caching optimization.

JIAN WANG received the B.S., M.S., and Ph.D. degrees from the College of Computer Science and Technology, Jilin University, China, in 2004, 2007, and 2011, respectively. He is currently a Professor with Jilin University. He has published over 50 articles on international journals. He is interested in topics related to wireless communication and vehicular networks, especially for network security and privacy protection.

HUI WANG received the B.S., M.S, and Ph.D. degrees from the College of Computer Science and Technology, Jilin University, in 2004, 2007, and 2010, respectively. She is currently with the Changchun University of Technology. She is interested in topics related to wireless networks and machine learning.

YUMING GE received the Ph.D. degree from the Institute of Computing Technology, Chinese Academy, in 2014. He is currently a Senior Engineer with the China Academy of Information and Communications Technology. He is interested in topics related to vehicular networks.