2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)

Novel Resource Allocation Algorithm of Edge Computing Based on Deep Reinforcement Learning Mechanism

Degan Zhang
Tianjin Key Lab of Intelligent Computing and Novel Software Technology, Key Laboratory of Computer Vision and System (Ministry of Education) Tianjin University of Technology
Tianjin, China
Email: 1751741712@qq.com

Hongrui Fan
Tianjin Key Lab of Intelligent Computing and Novel Software Technology, Key Laboratory of Computer Vision and System (Ministry of Education) Tianjin University of Technology
Tianjin, China
Email: 1357008606@qq.com

Jie Zhang
School of Electronic and Information Engineering Beijing Jiaotong University
Beijing, China
Email: zhang_jie2021@126.com

***Abstract*—Edge computing is a computing paradigm that can bring practical value to most modern enterprises. When it is integrated into the Internet of Things system, it can improve the QoS of mobile applications, and can realize real-time management of the generated big data.5G promotes the development of the edge computing paradigm further, and mobile users can obtain low-latency and high-speed access QoS. In this paper, we study the resource allocation method of edge servers in the MEC environment. We are absorbed in solving the problem of allocation of limited resources in the MEC system, such as computing resources and available bandwidth, with the goal of maximizing the average resource utilization and task processing capacity of edge servers in the MEC system, while considering some differential processing for delay-sensitive applications, describe this resource allocation problem as a Finite-state Markov Decision Process (FMDP), and consider the continuity of the user state, and design a new Edge Computing Resource Allocation Algorithm based on Deep Deterministic Policy Gradient (ECRAA-DDPG)to find the optimal strategy for resource allocation. Finally, a large number of experiments are used to prove the performance of the new algorithm, and the experimental results show that the method can make the optimal decision in a real environment.**

Keywords—computing paradigm, MEC, resource allocation, deep reinforcement learning

I. INTRODUCTION

Edge computing can be called a promising computing paradigm because it can accelerate almost all mainstream mobile applications (such as facial recognition, etc.). In realistic applications, edge computing is generally integrated into the IoT platform. In an edge computing environment, a base station is equipped with a certain amount of computing resources and can provide computing services for mobile users within the service range. When a large amount of data that needs to be processed in the IoT device, in order to ensure low latency, the great quantity of data generated can be offloaded to the "edge" of the network for computing processing[1-12]. The edge computing paradigm is not to process data in the long-range central cloud, instead, it makes full use of locality to bring data storage and computing resource closer to the equipment or data source that needs it most [13-21]. If edge computing and 5G technology are integrated, many applications will exhibit unprecedented low latency and high access speed. Through this paradigm in the Internet of Things, for tasks that are delay-sensitive or have a long waiting time, edge applications will significantly improve when executing tasks, making some applications that perform poorly (Such as online games, etc.) on traditional cloud platforms feasible.

Recently, Mobile Edge Computing has been introduced to improve the QoS of IoT systems[23-35]. This technology deploys computing resources closer to user devices, and can provide computing and storage services in the Radio Access Network (RAN) adjacent to the user devices. Some applications and services of the central cloud can be run in the MEC server, which not only greatly shortens the service delay of the system, but also reduces the burden of the Backhaul. In a certain sense, it can also improve the security of the IoT system.

Although the communication delay of MEC platform is significantly shorter than that of cloud computing platform, the computing capability, available bandwidth and other resources of its edge server are a little lower than those of the latter. Due to the constraints of limited resources, on the one hand, it is judged from the perspective of users, when many users request allocation, the close edge server cannot provide services for all users, and many tasks will be queued in the server, which will reduce QoS; on the other hand, judging from the perspective of edge servers, there is no superior resource allocation strategy (such as users requesting too many or too few resources, and some servers are idle for a long time) will waste a lot of edge computing platform energy[36-40]. Therefore, it is necessary to propose a reasonable resource allocation strategy for the edge computing platform to improve the resource utilization of the server.

Consequently, we regard the dynamic allocation of server resources in the edge computing platform as an important challenge. In this paper, we are devoted to improving the resource utilization of edge servers and reducing server energy consumption. In response to the above challenges, we propose a new resource allocation method of edge computing based on deep reinforcement learning mechanism. The main contributions of this paper are as follows:

- Determining the offloading server for each user belongs to a discrete variable problem, and the allocation of resources is a continuous variable problem. We group

978-0-7381-2646-3/21/$31.00 ©2021 IEEE
DOI 10.1109/ISPA-BDCloud-SocialCom-SustainCom52081.2021.00067

the users' states, so we describe this problem as a FMDP.

- We have formulate state information, behavior information, and reward function in the strategy. In addition, in order to make the strategy performance closer to the real environment, we also designed a series of continuous action, and thus proposed ECRAA-DDPG.
- In this method, in order to avoid a local optimal strategy in the system, we add an experience replay to the method so that the method can find the global optimal strategy as soon as possible.

II. Related Works

In the MEC system, a reasonable resource allocation method can effectively improve the resource utilization of the whole edge servers and reduce unnecessary energy consumption in the system. Now the academic and industrial circles have made a lot of innovation and improvement on the resource allocation method of MEC system. In [41], by solving the problem of edge server placement to overcome the limited resources and bandwidth bottlenecks of the device, firstly, the k-means algorithm was used to group the edge servers to balance the workload of MEC system, and then the mixed integer quadratic programming algorithm was used to further optimize the specific location of the server. Q. Peng et al. [42] combined user mobility and proposed an online decision allocation algorithm that supports mobile perception and dynamic migration to improve user resource utilization, but only a single application was considered in the experiment. The competition between multiple applications was not considered. You et al. [43] described the optimal resource allocation method as a non-convex mixed integer problem in the MEC platform, and defined it as an average offload priority function, thus proposing an energy-saving resource allocation scheme.

For problems that require continuous decision-making, Markov decision process (MDP) can be used as a representative theory to describe such problems. Reinforcement learning mechanisms can be used to solve the stochastic optimization difficulties described as MDP problems. In an environment that is difficult for us to predict, the agent of reinforcement learning can interact with the future environment and continuously learn to find the best strategy [44]. In [22], the computational offloading problem of multiple IOT devices was established as an evolutionary game model, analyzed the evolution process of IoT devices using replication dynamics, and designed an evolutionary game algorithm based on reinforcement learning. In [45], a reinforcement learning framework was used to optimize the computational offloading and resource allocation problems, Liu et al. proposed the ε - greedy Q learning method to minimize the weighted cost of delay and energy. In [46], the Double-Dueling DQN algorithm is proposed to provide resource allocation strategies in the Internet of Vehicles system, including network optimization, cache and computing resources. In [47], focuses on reducing the cost of task scheduling delay in edge computing networks, and uses Asynchronous Advantage Actor-Critic strategy (CECS-A3C) to optimize the scheduling difficulties of edge collaboration.

III. Model Description

A. System Model

In this paper, we are considering an in-vehicle network scenario with a single cellular network, as shown in left side of Fig. 1 shows the network topology in the case of vehicle-to-vehicle (V2V) communication at the intersection, and right side of Fig. 1 shows a vehicle-to-infrastructure (V2I) communication scene when vehicle are driving on a straight road. The system is composed of vehicle nodes, Road Side Units and MEC servers. Among them, the vehicle node mainly has three modules: on-board unit, GPS and wireless communication. The vehicle-mounted unit mainly executes some simple computing tasks, the GPS can share the location of the vehicle node, and the vehicle node communicates with nearby vehicles or roadside units through wireless communication. The Road Side Unit is arranged on the roadside and is mainly used to communicate with the vehicle-mounted unit.

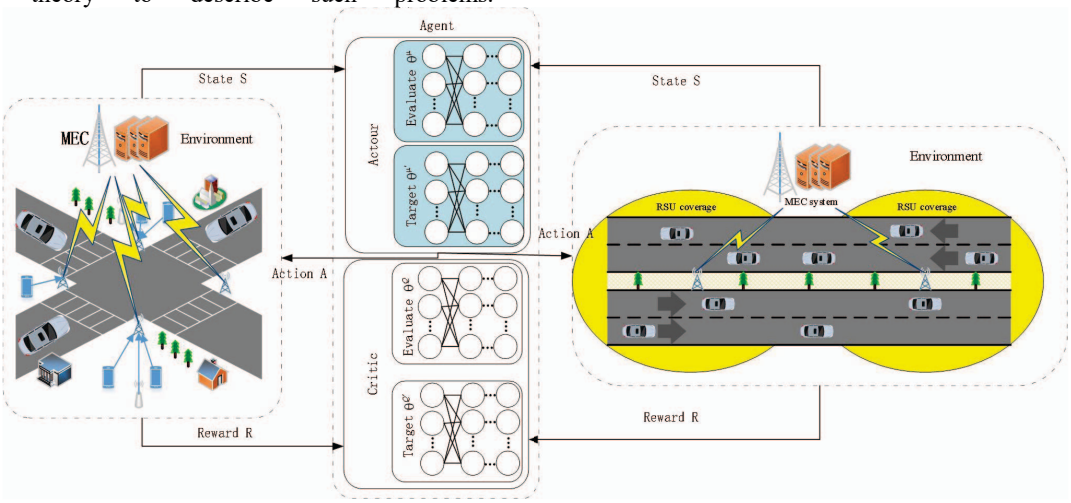


Fig. 1. System model example

What we consider is the multi-edge server resource allocation scenario of multi-user nodes, $M=\{1,2,3,...,m\}$ User Devices (UDs), $E=\{1,2,3,...,e\}$ edge server nodes, These servers are equipped with limited bandwidth $B(Hz)$ and have limited parallel computing capabilities $F(Hz)$, $N=\{1,2,3,...,n\}$ channels. The user equipment is only equipped with a single cellular network, so the user equipment can only connect to one channel at the same time. The user devices using the same channel may cause inter-channel interference, and each user device can only select one MEC server to perform tasks at the same time.

Suppose there are $K=\{1,2,3,...,k\}$ heterogeneous tasks; each task $k\in K$ has two parameters, $d_k(Hz)$ represents the computing resources to be requested by this task, $b_k(byte)$ indicates how much data needs to be entered. The information of the user device m connecting to the edge server e at time t can be expressed as:

$$Q_m^t=\begin{cases}0, & \text{if user } m \text{ is not allocated in } t\\ (d_k,b_k) & \text{if uesr } m \text{ connects to server } e \text{ in } t\end{cases} \tag{1}$$

Y_{me}^t represents the Boolean variable of whether the user device m is connected to the edge server e at time t , the formula is as follows:

$$Y_{me}^t=\begin{cases}0, & \text{if user } m \text{ is not in } t\\ 1, & \text{otherwise}\end{cases} \tag{2}$$

Some computing tasks have higher popularity (such as rendering scenes in VR), these tasks will be requested multiple times and will be run multiple times, so we assume that the number of user devices exceeds the number of tasks ($m\geq k$) . $k_{t,m}\in K$ represents the m-th UDs requesting tasks at time t , $k_t=[k_{t,1},...,k_{t,N}]^T$ represents the task request vector of all user devices. The popularity $p_{k,t}$ of each task follows the Zipf distribution, task popularity vector model $p_t=[p_{1,t},...,p_{k,t}]^T$, suppose the popularity $z_{k,t}\in K$ of task k at time t , the corresponding task popularity can be expressed as:

$$\phi_{\mathrm{k,t}}=\frac{z_{k,t}^{-\eta}}{\sum_{l=1}^{K}z_{l,t}^{-\eta}} \tag{3}$$

Among them, $\eta\geq 0$ in the Zipf distribution can show the popularity of the task. When $\eta=0$, it means that all tasks have the same popularity. The greater the η difference between tasks, the greater the difference between the popularity of tasks.

MEC platform computing resource allocation vector $f_t=[f_{1,t},...,f_{m,t}]^T$, the total allocation of computing resources cannot exceed the computing resources of the edge server, namely:

$$\sum_{i=1}^{m}1(Y_{ie}^t\neq 0\vee k_{t,i}\in k_t)f_{i,t}\leq F,\quad \forall t\in T \tag{4}$$

Among them, " $\vee$ " stands for logical "or" operation. T represents the collection from the first to the now. $1(x)$ is an indicator function, if the event is true, then $1(x)=1$, otherwise it is 0.

Therefore, the resource utilization of edge server e before the time t is expressed as follows:

$$\eta_e^T=\sum_{t=0}^{T}\frac{\sum_{m=0}^{m}Y_{me}^t}{f_{m,t}} \tag{5}$$

Definition 1: C_e represents the energy consumption of all edge servers to execute computing tasks, and C_e mainly includes two aspects: a) energy consumption CL_{e} for edge server operation; Data transfer energy consumption CM_e between UDs and edge servers, here we only consider the energy consumption of the device to transmit data to the edge server without considering the energy consumption of the device to receive the returned data.

Execution energy consumption CL_{e} of all edge servers is as follows:

$$CL_{\mathrm{e}}=\sum_{t\in T}\sum_{e\in E}\delta_{\mathrm{e}}f_t \tag{6}$$

Where δ_{e} represents the energy consumption coefficient of each CPU cycle, which can be obtained by the method in [48].

Data transmission energy consumption CM_e between user equipment and edge server is as follows:

$$CM_e=P_e\frac{b_k}{B\ \log_2(1+\frac{P_eG_{e,n}}{\partial^2+\sum_{i=1,i\neq e}^{e}P_iG_{i,n}})} \tag{7}$$

Where B represents the bandwidth of the channel, P_e represents the transmission power, $G_{e,n}$ represents the channel gain, ∂^2 represents the thermal noise power of the channel, and $\sum_{i=1,i\neq e}^{e}P_iG_{i,n}$ represents the interference of other devices on the same channel.

Therefore, we can conclude that the total energy consumption of edge servers in the MEC platform is:

$$C_e=\sum_{t\in T}\sum_{e\in E}\delta_e f_t+\sum_{e\in E}\sum_{m\in M}Y_{me}^tP_e\frac{b_k}{B\ \log_2(1+\frac{P_eG_{e,n}}{\partial^2+\sum_{i=1,i\neq e}^{e}P_iG_{i,n}})} \tag{8}$$

We aim to maximize the average resource utilization and task processing capacity of edge servers in the MEC system, and the optimal allocation strategy O_1 can be expressed as follows:

$$\begin{aligned} & O_1: MAX(\frac{\sum_{j=1}^{e} \eta_j^T}{e}, -C_e) \\ & s.t. C1: UDm, n \in \{0,1\}, \\ & \quad C2: (4) \\ & \quad C3: priority(\phi_{k1,t}) > priority(\phi_{k2,t}) \phi_{k1,t} > \phi_{k2,t} \\ & \quad C4: \forall m \in M, \forall n \in N \end{aligned} \tag{9}$$

Among them, the constraint $C1$ represents that each user can connect to at most one channel at a time of selection. Constraint $C2$ means that the total computing resources allocated cannot exceed the computing resources of the edge server. Constraint $C3$ means that task k_1 is executed first than k_2.

It is worth mentioning that in the real MEC environment, describing the complete mathematical model of O_1 is not an optimal solution. In addition, O_1 is a mixed-integer nonlinear programming, and it is difficult to solve with a large number of statistical distributions. An online strategy can be designed to solve this type of problem, allowing the environment and the system to interact in real time to implement a resource allocation plan. Therefore, we propose a new method based on deep reinforcement learning mechanism to find the optimal solution, instead of the traditional optimization method to solve the NP-hard (Non-deterministic Polynomial-hard) problem.

B. Problem Transformation

The above problems have Markov properties, that is, they can replace the current state after implicit evolution in the future, and the historical state and the current state are relatively independent, so we describe it as FMDP. A typical DFMDP is represented by a tuple (S, A, p, r), where S is a small number of state spaces, A is a small number of action spaces, p is the transition probability from state $s(\forall s \in S)$ to state $s'(\forall s' \in S)$ after performing action $a(\forall a \in A)$, and r is the real-time reward obtained after performing action $a(\forall a \in A)$. In this paper, we need to find a deterministic strategy μ, which can map a state to a specified action, namely $\mu: S \to A$.

We regard the edge server as the environment, and the level management of the edge server plays the role of an agent. This agent continuously executes decisions and interacts with the environment. In the resource allocation of MEC discussed in this paper, the states message of the task requested by the user is regarded as the states, and regard scheduling tasks or adjusting computing resources as actions, which are executed by the agent based on policy. The action of the agent generates reward feedback in the environment. The agent select policy to enter a new state according to the current state, reward and the environment. This policy is in principle to increase the probability of the agent getting a higher reward. Here we have designed the reward feedback to be consistent with the optimization goal. For the system in this study, we define the state space, action space, state transition and reward functions respectively:

1) State space

Definition 2: State $s \in S$ describes the state information of mobile edge applications deployed on the MEC system, expressed as follows:

$$S \triangleq \{s \mid s = (K, N, Y_{me}^t)\} \tag{10}$$

Among them, K is the set of all tasks that need to be processed, and N is the set of all channels. Y_{me}^t represents the connection status between device m and the edge server. Because of the difference of user devices, K and N in the state are uncertain. At each moment, channel N is estimated based on channel reciprocity to estimate the future uplink transmission, and K is directly transmitted to the edge server through the channel. Therefore, the dimension of the vector in the state space is $k + nm$.

Algorithm 1 Resource allocation state initialization

```
1:InPuts: Number of user equipment, number of edge servers, computing
resources of edge servers, etc.
2:for i=1 to E do
3:    Count edge server computing resources;
4: for j=1 to Total number of connections between edge servers do
5:    Statistics of available bandwidth;
6: for k=1 to M do
7:    Count the user's initial location and connection information
(disconnected by default);
8:Return state set s;
```

2) Action space

According to the current state s, the agent will choose action a on the basis of decision O_1, we have:

$$A \triangleq \{a \mid a = (O_1)\} \tag{11}$$

When the number of UDs is large, the action space will also become complicated. Therefore, the policy decision of the action space is the focus and difficulty of this problem.

3) State transition

This process is a mapping from a state to an action (S, A). When a state $s = (K, N, Y_{me}^t)$ executes decision O_1 at time t, an action $a \in A$ will be selected to transform the state into a subsequent state $s' = (K', N', Y_{me}^{t\prime})$, that is, $s \xrightarrow{a} s'$.

4) Reward function

When the agent takes an action a by observing a specific state s, it will immediately get a reward signal r. Our goal is to improve the resource utilization of edge servers and reduce server energy consumption. Therefore, the reward function r_t is defined as:

$$r_t = R(s_t, a_t)$$
$$= \omega_\eta \eta_e^t - \omega_c \left(\sum_{e\in E} \delta_e f_t + \sum_{e\in E}\sum_{m\in M} Y_{me}^t P_e \frac{b_k}{B\log_2\left(1+\frac{P_e G_{e,n}}{\partial^2 + \sum_{i=1,i\neq e}^{e} P_i G_{i,n}}\right)}\right) \quad (12)$$

Where ω_η represents the weight parameter of resource utilization in the system, and ω_c represents the weight parameter of energy consumption in the system.

IV. Design and Implementation of Algorithm

According to strategy O_1 and formula 11, it is confirmed that the action space A is continuous and discretized, and is a high-dimensional state space. Therefore, we use Deep Reinforcement Learning (DRL) to solve this problem. DRL can be seen as a combination of Deep Neural Network (DNN) and Reinforcement Learning. The Deep Deterministic Policy Gradient (DDPG) algorithm uses a combination of policy gradient and DQN (Deep Q-Learning) advantages [49], which can solve discrete and continuous problems. In addition, DDPG updates the model weights at each step, which means that the algorithm can immediately adapt to the dynamic environment. Therefore, we can use the DDPG framework to find the best strategy for this research.

A. Algorithm description

As shown in Fig. 1, the agent in DDPG consists of an actor network and a critic network, and both of them are implemented by two DNNs. The two DNNs are the target network and the evaluation network. For the state input from the environment, the actor network makes an action decision, and the critic network uses a Q function to evaluate each set of state-action mappings. The standard Q function is expressed as follows:

$$Q(s,a) = \mathbb{E}\left[\sum_{\tau=0}^{\infty} \gamma^\tau r(t+\tau) | \pi, s = s(t), a = a(t)\right] \quad (13)$$

Among them, $r(t)$ represents the real-time reward obtained by the agent at time t, which is expressed as $r(t) = r_t$ in this study; γ is the discount factor in $r(t)$.

The parameter vector θ in the Actor network $\mu(s;\theta^\mu)$ function is jointly determined by the actor network and the critic network. In this study, it is expressed as a specific strategy from a certain state to a certain action. The Actor network starts from the J-distribution and uses the chain rule to obtain the expected benefits to update the network.

Lemma 1: Assuming that $A.1$ is satisfied in the MDP (the relevant variables are continuous under the parameters S, A, s', appendix in [49]), then $\nabla_{\theta^\mu} Q(s,a;\theta^Q)$ and $\nabla_{\theta^\mu}\mu(s;\theta^\mu)$ exists, and the deterministic policy gradient exists, which is:

$$\nabla_{\theta^\mu} J \approx \mathbb{E}\left[\nabla_{\theta^\mu} Q(s,a;\theta^Q)|_{s=s^k, a=\mu(s^k|\theta^\mu)}\right]$$
$$\approx \mathbb{E}\left[\nabla_{\theta^\mu} Q(s,a;\theta^Q)|_{s=s^k, a=\mu(s^k)} \nabla_{\theta^\mu}\mu(s;\theta^\mu)|_{s=s^k}\right] \quad (14)$$

The proof of the above lemma is given in the appendix [49]. At the same time, the update of the Actor network is based on the above lemma.

Lemma 2: If the approximate function $Q(s,a;\theta^Q)$ matches a deterministic strategy $\mu(s;\theta^\mu)$, that is $\nabla_{\theta^\mu} J = \mathbb{E}\left[\nabla_{\theta^\mu} Q(s,a;\theta^Q)|_{s=s^k, a=\mu(s^k)} \nabla_{\theta^\mu}\mu(s;\theta^\mu)|_{s=s^k}\right]$, then two conditions need to be met:

a) $\nabla_{\theta^\mu} Q(s,a;\theta^Q)|_{s=s^k, a=\mu(s^k)} = \nabla_{\theta^\mu}\mu(s;\theta^\mu)|_{s=s^k}{}^T \omega$

b) ω is to minimize the mean square error, $MSE(\theta,\omega) = \mathbb{E}\left[(s;\theta,w)^T (s;\theta,\omega)\right]$, among them, $\epsilon(s;\theta,\omega) = \nabla_{\theta^\mu} Q(s,a;\theta^Q)|_{s=s^k, a=\mu(s^k)} - \nabla_{\theta^\mu} Q(s,a;\theta^\mu)|_{s=s^k, a=\mu(s^k)}$.

The above lemma is given in [49].

It is worth mentioning that exploration is an important challenge for learning in the continuous action space, which requires that the target value should be updated slowly to increase learning stability. Therefore, the target network in the actor network and the critic network cannot be updated too fast with the learning network.

B. Algorithm design

In summary, we propose a new resource allocation method of edge computing based on deep reinforcement learning mechanism, the method is as follows:

Algorithm 2 ECRAA-DDPG

1 : InPuts:Relevant parameters in the system model, namely, the number of user devices, the number of edge servers, the computing resources of edge servers, etc.
2 : ECRAA-DDPG related parameters：The number of rounds D_{max}, The time step T_{max} in each round, the same playback buffer $\mathcal{R}_B$.
3 : for i in S do
4 : Randomly initialize the weight θ_s^Q of critic network $Q_s(S^s, a^s | \theta_s^Q)$;
5 : Randomly initialize the weight θ_s^μ of the actor network $\mu_s(S^s | \theta_s^\mu)$;
6 : Initialize the weight $\theta_s^{Q\prime} \leftarrow \theta_s^Q$ of the target network Q_s';
7 : Initialize the weight $\theta_s^{\mu\prime} \leftarrow \theta_s^\mu$ of target network μ_s';
8 : Initialize the playback buffer $\mathcal{R}_B^s(\forall s \in S)$ is an empty array;
9 : end for
10: for j=1 to D_{max} do
11: According to Algorithm 1, initialize the status information for each edge task $S_1 \triangleq \left\{S_1^s\right\}_{s\in S}$;
12: for x=1 to T_{max} do
13: The edge servers in the MEC system share task information;
14: for i=1 to S do
15: Find the optimal solution for each s according to formulas (10)-(14);
16: end for
17: end for
18: end for
19: Output: Optimal strategy μ_s

C. Algorithm complexity analysis

In this section, We will discuss the space complexity of the ECRAA-DDPG algorithm. In this method, we use the S representation of state space and the A representation of action space. Therefore, the space complexity of ECRAA-DDPG algorithm is $O(|S||A|)$.

V. Simulation Results

In order to verify the performance of our proposed algorithm, we used Google Tensorflow-2.0.0 to conduct a lot of experiments, and introduced user mobility data collected by mobile devices at Seoul Metro Station in South Korea provided by CRAWDAD, several main parameters of the algorithm are analyzed for robustness. We compared the proposed ECRAA-DDPG algorithm with other four baseline strategies, namely DQN-based decision-making, PCL (Popularity-based Caching and Local execution), PCO (Popularity-based Caching and full Offloading), RCOR (Randomized Caching, Offloading, and Resource allocation).

A. Parameter setting of real scenario

As shown in Fig. 1, a real vehicle network scenario is established. The specific parameters are shown in Table I and Table II.

TABLE I. Experimental Parameters

Parameter	Value
Number of vehicles	{5，7，9，11}
Task queue	10
Number of channels	8
Number of edge servers	{10，15，20，25，30}
Channel bandwidth	100MHz
Transmission power	0.5W
Computing resources of edge servers	{3,5,7,9,11,13}GHz
Vehicle computing resources	0.5GHz
Waiting time	{20，50}ms
Task node size	[0.2，1]MB
Delay threshold	{10，40，100}ms
Energy Density	1.25x10-26J/Cycle
Thermal noise power of the channel	70dBm
Server coverage	500m

TABLE II. ECRAA-DDPG Parameters

Parameter	Value
Number of steps per episode	100
Experience buffer pool	20000
Critic network learning rate	0.001
Actor network learning rate	0.0001

B. Experimental results and analysis

We first focus is on the performance of the ECRAA-DDPG algorithm. In Fig. 2, we show the convergence performance of our proposed ECRAA-DDPG algorithm under unequal number of users. As can be seen from the curve in the figure, as the number of episodes increases, the overall reward of the system gradually increases, and basically maintains a stable reward after 100 to 200 rounds. From the comparison of these two sets of curves, it can be seen that in the initial learning phase, when there are many users, the ECRAA-DDPG algorithm converges more slowly than when there are few users, because the increase in the number of users will increase the dimensions of the state space and the action space. We proposed ECRAA-DDPG algorithm needs some time to explore the optimal strategy.

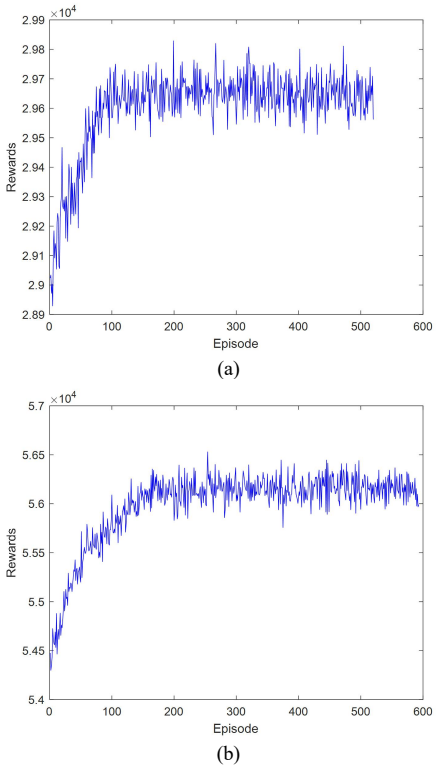


Fig. 2. Algorithm convergence (a) 15 users (b) 30 users

Second, Fig. 3 compares the influence of different computing resources of edge servers on the average energy consumption in the system in several methods. It can be seen from the figure that with the increase of computing power, the average energy consumption tends to decrease to varying degrees. PCL has the least downward trend, because this method manages the fewest user equipment in the MEC platform. The ECRAA-DDPG algorithm we proposed can achieve the optimal performance very well. In fact, with the increase of edge server computing resources, both the ECRAA-DDPG method and the DQN method can improve the system energy consumption, but the DQN method will generate a high-dimensional action space, making the algorithm more complex.

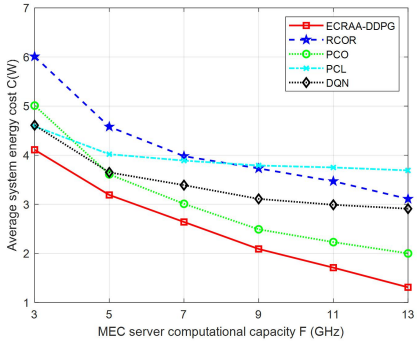
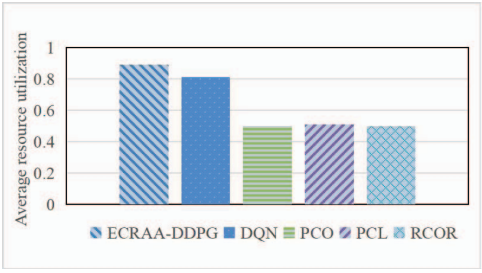
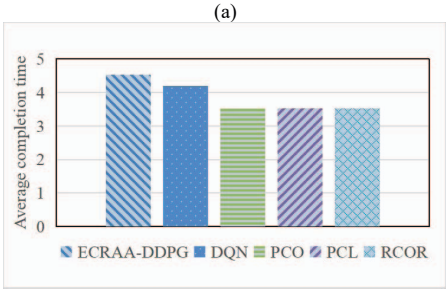


Fig. 3. Average energy consumption of different computing resources

Finally, we compared the proposed ECRAA-DDPG algorithm with the other four baseline strategies from multiple perspectives. From Figure 4(a), we can conclude that the decision made by the ECRAA-DDPG algorithm saves about half of the resources compared to the RCOR decision. Just because the requested fewer resources are enough to complete the current task, subsequent tasks are in a short queue. As shown in Figure 4(b), the average completion time of each task of the ECRAA-DDPG algorithm is longer, and this side effect caused by system weighting is minimal and acceptable for most IOT edge tasks.



(a)



(b)

Fig. 4. (a) Average resource utilization (b) Average completion time of a single task

VI. Conclusion

This paper proposes a new resource allocation method of edge computing based on deep reinforcement learning mechanism. According to the characteristics of the resource allocation problem in edge computing, we describe it as a FMDP. Based on the idea of deep reinforcement learning, the DDPG framework is used to achieve the goal of maximizing the average resource utilization and task processing capacity of the edge servers in the MEC system, and to avoid local optimization in the system, we have added an experience pool to the algorithm. In addition, we have proved the good performance of the proposed ECRAA-DDPG method through a large number of experiments.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (No.61571328), Tianjin Key Natural Science Foundation (No.18JCZDJC96800).

References

[1] T. ZHANG, "Novel Self-Adaptive Routing Service Algorithm for Application of VANET," Applied Intelligence, 2019, vol. 49, no. 5, pp. 1866-1879.

[2] X. WANG and X. D. SONG, "A Novel Approach to Mapped Correlation of ID for RFID Anti-collision," IEEE Transactions on Services Computing, 2014, vol. 7, no. 4, pp. 741-748.

[3] J. N. YANG and G. Q. MAO, "Optimal Base Station Antenna Downtilt in Downlink Cellular Networks," IEEE Transactions on Wireless Communications, 2019, vol. 18, no. 3, pp. 1779-1791.

[4] J. X. GAO, "Novel Approach of Distributed & Adaptive Trust Metrics for MANET," Wireless Networks, 2019, vol. 25, no. 6, pp. 3587–3603.

[5] D. G. ZHANG, G. LI and K. ZHENG, "An energy-balanced routing method based on forward-aware factor for Wireless Sensor Network," IEEE Transactions on Industrial Informatics, 2014, vol. 10, no. 1, pp. 766-773.

[6] S. LIU, "Novel Unequal Clustering Routing Protocol Considering Energy Balancing Based on Network Partition & Distance for Mobile Education," Journal of Network and Computer Applications, 2017, vol. 88, no. 15, pp. 1-9.

[7] T. ZHANG, "Novel Optimized Link State Routing Protocol Based on Quantum Genetic Strategy for Mobile Learning," Journal of Network and Computer Applications, 2018, vol. 122, pp. 37-49.

[8] H. GE, "New Multi-hop Clustering Algorithm for Vehicular Ad Hoc Networks," IEEE Transactions on Intelligent Transportation Systems, 2019, vol. 20, no. 4, pp. 1517-1530.

[9] J. Q. CHEN and G. Q. MAO, "Capacity of Cooperative Vehicular Networks with Infrastructure Support: Multi-user Case," IEEE Transactions on Vehicular Technology, 2018, vol. 67, no. 2, pp. 1546-1560.

[10] T. ZHANG and J. ZHANG, "A Kind of Effective Data Aggregating Method Based on Compressive Sensing for Wireless Sensor Network," EURASIP Journal on Wireless Communications and Networking, 2018, vol. 159, pp. 1-15. DOI: 10.1186/s13638-018-1176-4

[11] T. ZHANG, "A New Method of Data Missing Estimation with FNN-Based Tensor Heterogeneous Ensemble Learning for Internet of Vehicle," Neurocomputing, 2021, vol. 420, no. 1, pp. 98-110.

[12] L. CHEN and J. ZHANG, "A multi-path routing protocol based on link lifetime and energy consumption prediction for mobile edge computing," IEEE Access, 2020, vol. 8, no. 1, pp. 69058-69071.

[13] C. CHEN and Y. Y. CUI, "New Method of Energy Efficient Subcarrier Allocation Based on Evolutionary Game Theory," Mobile Networks and Applications, 2021, vol. 26, no. 2, pp. 523-536.

[14] Y. N. ZHU, "A new constructing approach for a weighted topology of wireless sensor networks based on local-world theory for the Internet of Things (IOT)," Computers & Mathematics with Applications, 2012, vol. 64, no. 5, pp. 1044-1055.

[15] S. ZHOU, "A low duty cycle efficient MAC protocol based on self-adaption and predictive strategy," Mobile Networks & Applications, 2018, vol. 23, no. 4, pp. 828-839.

[16] H. L. NIU, "Novel PEECR-based Clustering Routing Approach," Soft Computing, 2017, vol. 21, no. 24, pp. 7313-7323.

[17] K. Zheng and T. Zhang, "A Novel Multicast Routing Method with Minimum Transmission for WSN of Cloud Computing Service," Soft Computing, 2015, vol. 19, no. 7, pp. 1817-1827.

[18] T. ZHANG, "A Kind of Novel Method of Power Allocation with Limited Cross-tier Interference for CRN," IEEE Access, 2019, vol. 7, no. 1, pp. 82571-82583.

[19] X. H. LIU, "A New Algorithm of the Best Path Selection based on Machine Learning," IEEE Access, 2019, vol. 7, no. 1, pp. 126913-126928.

[20] P. Z. ZHAO and Y. Y. CUI, "A New Method of Mobile Ad Hoc Network Routing Based on Greed Forwarding Improvement Strategy," IEEE Access, 2019, vol. 7, no. 1, pp. 158514-158524.

[21] S. LIU, "Adaptive Repair Algorithm for TORA Routing Protocol based on Flood Control Strategy," Computer Communications, 2020, vol. 151, no. 1, pp. 437-448.

[22] Y. Y. CUI, "Novel Method of Mobile Edge Computation Offloading Based on Evolutionary Game Strategy for IoT Devices," AEU-International Journal of Electronics and Communications, 2020, vol. 118, no. 5, pp. 1-13.

[23] M. J. PIAO and T. ZHANG, "New Algorithm of Multi-Strategy Channel Allocation for Edge Computing," AEUE - International Journal

[24] J. X. WANG J X and H. R. Fan, "New Method of Traffic Flow Forecasting Based on Quantum Particle Swarm Optimization Strategy for Intelligent Transportation System," International Journal of Communication Systems, 2020, vol. 33, no. 10, pp. 1-13.

[25] X. H. Liu, "Novel Best Path Selection Approach Based on Hybrid Improved A* Algorithm and Reinforcement Learning," Applied Intelligence, 2021, vol. 51, no. 9, pp. 1–15.

[26] S. LIU, "Novel Dynamic Source Routing Protocol (DSR) Based on Genetic Algorithm-Bacterial Foraging Optimization (GA-BFO)," International Journal of Communication Systems, 2018, vol. 31, no. 18, pp. 1-20.

[27] C. P. ZHAO, "A new medium access control protocol based on perceived data reliability and spatial correlation in wireless sensor network," Computers & Electrical Engineering.2012, vol.38, no.3, pp.694-702.

[28] J. Q. CHEN, "A Topological Approach to Secure Message Dissemination in Vehicular Networks," IEEE Transactions on Intelligent Transportation Systems, 2020, vol.21, no.1, pp.135-148. DOI:10.1109/ TITS.2018.2889746

[29] C. L. GONG, K. W. JIANG, "A Kind of New Method of Intelligent Trust Engineering Metrics (ITEM) for Application of Mobile Ad Hoc Network," Engineering Computations, 2019, vol.37, no.5, pp.1617-1643. DOI: 10.1108/EC-12-2018-0579

[30] H. WU, P. Z. ZHAO, "New Approach of Multi-path Reliable Transmission for Marginal Wireless Sensor Network," Wireless Networks, 2020, vol.26, no.2, pp.1503–1517. DOI: 10.1007/s11276-019-02216-y

[31] P. B. DUAN, "A Unified Spatio-temporal Model for Short-term Traffic Flow Prediction," IEEE Transactions on Intelligent Transportation Systems, 2019, vol.20, no.9, pp.3212-3223. DOI:10.1109/TITS.2018.28 73137

[32] Y. M. TANG, "Novel Reliable Routing Method for Engineering of Internet of Vehicles Based on Graph Theory," Engineering Computations, 2019, vol.36, no.1, pp.226-247.

[33] X. WANG, X. D. SONG, "New Clustering Routing Method Based on PECE for WSN," EURASIP Journal on Wireless Communications and Networking, 2015, vol.2015, no.162, pp.1-13. DOI: 10.1186/s13638-015-0399-x

[34] K. ZHENG, D. X. ZHAO, "Novel Quick Start (QS) Method for Optimization of TCP," Wireless Networks, 2016, vol.22, no.1, pp.211-222.

[35] X. D. ZHANG, "Design and implementation of embedded un-interruptible power supply system (EUPSS) for web-based mobile application," Enterprise Information Systems, 2012, vol.6, no.4, pp.473-489

[36] D. G. ZHANG, "A new approach and system for attentive mobile learning based on seamless migration," Applied Intelligence, 2012, vol.36, no.1, pp.75-89.

[37] X. WANG, X. D. SONG, "New Medical Image Fusion Approach with Coding Based on SCD in Wireless Sensor Network," Journal of Electrical Engineering & Technology, 2015, vol.10, no.6, pp.2384-2392.

[38] S. LIU, "Dynamic Analysis For The Average Shortest Path Length of Mobile Ad Hoc Networks under Random Failure Scenarios," IEEE Access, 2019, vol.1, no.7, pp.21343-21358. DOI:10.1109/ACCESS. 2019.2896699

[39] X. H. Liu, "A path planning method based on the particle swarm optimization trained fuzzy neural network algorithm," Cluster Computing, 2021, vol.24, no.1, pp.1-15. DOI:10.1007/s10586-021-03235-1

[40] C. L. Gong, "A new algorithm of clustering AODV based on edge computing strategy in IOV," Wireless Networks, 2021, vol.27, no.4, pp.2891-2908. DOI:10.1007/s11276-021-02624-z

[41] G. YAN, "User allocation-aware edge cloud placement in mobile edge computing," Software: Practice and Experience, 2020, vol.50, no.5. DOI: 10.1002/spe.2685.

[42] Q. L. PENG, "Mobility-Aware and Migration-Enabled Online Edge User Allocation in Mobile Edge Computing," 2019 IEEE International Conference on Web Services (ICWS), 2019, vol.1, no.1, pp.91-98. DOI: 10.1109 /ICWS.2019.00026.

[43] C. S. YOU, "Energy-effificient resource allocation for mobile-edge computation offloading," IEEE Trans Wireless Commun, 2017, vol.16, no.3, pp.1397–1411. DOI: 10.1109/TWC.2016.2633522.

[44] Z. Y. LIU, "Deep Reinforcement Learning Based Dynamic Resource Allocation in 5G Ultra-Dense Networks," 2019 IEEE International Conference on Smart Internet of Things (SmartIoT), 2019, pp.168-174. DOI: 10.1109/SmartIoT.2019.00034.

[45] X. L. LIU, "Resource Allocation for Edge Computing in IoT Networks via Reinforcement Learning," ICC 2019 - 2019 IEEE International Conference on Communications (ICC), 2019, pp.1-6. DOI: 10.1109/ICC. 2019.8761385.

[46] Y. HE, " Integrated Networking, Caching, and Computing for Connected Vehicles: A Deep Reinforcement Learning Approach," IEEE Transactions on Vehicular Technology, 2018, vol.67, no.1, pp.44-55. DOI: 10.11 09/TVT.2017.2760281.

[47] Q. FAN, "Deep Reinforcement Learning Based Task Scheduling in Edge Computing Networks," 2020 IEEE/CIC International Conference on Communications in China (ICCC), 2020, vol.1, no.1, pp.835-840. DOI: 10.1109/ICCC49849.2020.9238937.

[48] Y. G. WEN, "Energy-optimal mobile application execution: Taming resource-poor mobile devices with cloud clones," Proc. IEEE INFOCOM, 2012, vol.1, no.1, pp.2716–2720. DOI: 10.1109/INFCOM. 2012. 6195685.

[49] S. DAVID, "Deterministic Policy Gradient Algorithms," ICML, 2014(6), vol.1, no.1, pp.1-8.