ISSN: 2688-8645

Offloading Strategy for Mobile Edge Computing based on Reinforcement Learning

Ji Zhang, Peng Dong*

School of Control and Computer Engineering, North China Electric Power University, Baoding 071003, China

*dongpeng9624@163.com

Abstract

The progress of science and technology makes intelligent mobile devices more and more popular, and mobile device traffic increases rapidly. However, due to its limited resources and computing performance, intelligent mobile devices may face insufficient capability in processing computation-intensive applications. Using mobile edge computing technology can provide high performance, low latency and high bandwidth carrier-class service environment, so as to effectively solve the problem. In order to minimize the total weighted cost of the system, a computational offload and resource allocation scheme based on reinforcement learning is proposed in this paper. Simulation results show that the proposed algorithm can effectively reduce the total weighted cost of the system compared with some benchmark methods.

Keywords

Mobile Edge Computing; Computation Offloading; Reinforcement Learning.

1. Introduction

With the vigorous development of science and technology, intelligent terminal has become an indispensable part of modern life. At present, the fifth generation of mobile technology (5G) is facing new challenges of the coexistence of explosive data traffic growth and massive device connections. At the same time, 5G network's new business scenarios, such as driverless cars, smart grids and industrial communications, also put forward higher requirements on indicators such as delay, energy efficiency, number of device connections and reliability. In order to cope with the rapid development of mobile Internet and Internet of Things, 5G needs to meet the new business requirements of ultra-low delay, ultra-low power consumption, ultra-high reliability and ultra-high density connection [1-2].

However, due to the limitation of computing capacity and battery capacity, terminal devices cannot efficiently meet the basic requirements of low latency and high computing for a large number of new services. Offloading computation-intensive tasks to the cloud increases transmission latency and additional network load. Mobile cloud computing allows mobile devices to partially or completely migrate local computing tasks to the cloud server for execution, thus solving the resource shortage problem of mobile devices and saving energy consumption for local task execution. However, offloading tasks to the cloud server located in the core network needs to consume back link resources, resulting in additional delay cost, which cannot meet the requirements of low delay and high reliability in 5G scenarios [3]. Mobile Edge Computing (MEC) was first proposed by The European Telecommunication Standards Association in 2014. MEC system allows devices to offload computing tasks to network edge nodes, such as base stations and wireless access points, which not only meets the expansion needs of terminal devices' computing capacity. At the same time, it makes up for the disadvantage of long delay of cloud computing [4]. MEC has quickly become a key technology

ISSN: 2688-8645

of 5G, helping to achieve key technical indicators such as ultra-low delay, ultra-high energy efficiency and ultra-high reliability of 5G services.

2. Research Status

In recent years, scholars at home and abroad have carried out in-depth research on MEC computational offloading technology. Literature [5] proposed a one-dimensional search algorithm to find the optimal strategy for computing offloading, which is suitable for the case that all computing tasks need to be offloaded. Simulation results show that the proposed method can effectively reduce the delay time by 80% compared with the local calculation of tasks on mobile devices. Literature [6] considered the application scenarios of the Internet of Things and proposed a completely polynomial time approximation scheme based on reasonable allocation of resources to programs to reduce computing delay. Experimental results show that lower computing delay can be obtained than heuristic algorithms. In literature [7], a distributed computing offloading algorithm is designed based on game theory, and the calculation delay index is quantified to achieve lower computing time cost. Literature [8] proposed off-line pre-computation offloading strategy, which solved the problem of minimizing computational offloading energy of mobile devices. Experimental results show that compared with local computation, it can effectively reduce energy consumption by 78%. Literature [9] proposed an efficient and energy saving computational offloading algorithm, which is suitable for the case of multi-mobile user devices carrying out all computational offloading. Simulation results show that this method can reduce the energy consumption of user equipment by 15%. Literature [10] solved the optimization problem of resource allocation through mutual iteration of The Chess algorithm and the Hungarian algorithm, so as to reduce the timely delay of computing energy consumption and minimize the total system cost. Literature [11] proposed a content-aware classification offloading strategy based on MEC, so that messages generated by vehicle terminals can be processed directly at edge nodes to ensure the reliability of security messages and the balance of delay and energy consumption. In reference [12], a offloading scheme based on game theory was proposed to solve the offloading problem of multi-user tasks, which made the optimal solution approximate to the theoretical optimal strategy and greatly reduced the system cost.

3. System Model

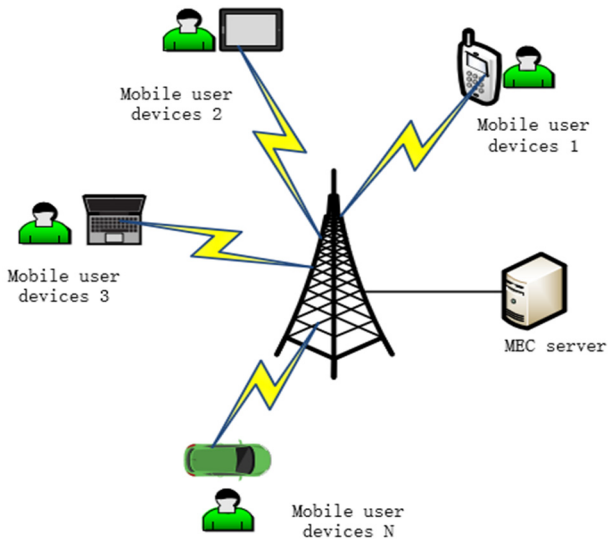

Figure 1. Single cell scenario model

ISSN: 2688-8645

As shown in Figure 1, a single cell scenario consisting of multiple mobile user devices and a single MEC server is considered. Among them, there is an eNB base station, which is deployed with a MEC server to provide computing services for multiple mobile user devices. Assume that each mobile user device has a computationally intensive task to complete and can offload the task to a MEC server over a wireless network or perform local computing.

3.1. Communication Model

The set of mobile user devices is N={1,2,3..., N}. The uninstallation decision variable of the mobile user device is defined as $a_n \in \{0,1\}(n \in N)$, where $a_{\mathrm{n}} = 1$ represents the mobile user device N chooses to uninstall to the MEC server for computing processing. $a_n = 0$ indicates the mobile user device. N Perform computing on the local terminal. Here, W represents the wireless channel bandwidth. If multiple users choose the offloading task at the same time, the wireless channel bandwidth will be evenly allocated to each user. Therefore, when N users choose to uninstall computing at the same time, the data upload rate is

$$r_n = \frac{W}{N}\log_2\left(1+\frac{P_{\mathrm{n}} g_n}{\frac{W}{N}N_0}\right) \tag{1}$$

p_n is the transmission power of the mobile user device when data is uploaded, g_n is the channel gain of the wireless channel allocated to the mobile user device, and N_0 is the variance of the Gaussian white noise channel during data transmission.

3.2. Calculation Model

Suppose every mobile user equipment have a computationally intensive task need to complete $R_n = \{B_n, D_{\mathrm{n},} T_n^{\max}\}$, B_n according to the size of the input data needed for computing tasks R_n, D_n said the total number of CPU cycles needed to complete the computing tasks, $T_n^{\max}$ says the biggest can tolerate delay computing tasks R_n, namely each mobile user equipment task delay no more than $T_n^{\max}$. The computing task processing modes under consideration include local computing and offloading computing, which are respectively introduced below.

3.2.1. Local Computing Model

If mobile user device n chooses to perform task R_n locally, the cost includes local execution delay T_{n}^l and energy consumption E_n^l. Definition f_n^l refers to the local computing power of mobile user devices, expressed in CPU cycles per second. The computing power of different mobile user devices is different. The execution delay of the local computing task is:

$$T_{\mathrm{n}}^l = \frac{D_n}{f_n^l} \tag{2}$$

The energy consumption for local execution is:

$$E_n^l = z_n (f_n^l)^2 D_n \tag{3}$$

The local execution cost weighted sum of mobile user devices is:

$$C_n^l = \theta_1 T_n^l + \theta_2 E_n^l \tag{4}$$

ISSN: 2688-8645

Where, $0 \le \theta_1, \theta_2 \le 1$ represents the weight parameters of execution delay and energy consumption of mobile users respectively.

3.2.2. Offloading Computing Model

If the mobile user device N chooses to perform computing tasks by offloading computing, the whole execution process includes three parts: First, the mobile user device needs to upload a large amount of data to the base station, and then the base station transfers the data to the MEC server; Secondly, MEC server allocates certain computing resources to perform computing tasks. Finally, the MEC server returns the results to the mobile user device.

According to the above procedure, the first part is the transmission delay caused by the transmission of input data, expressed as:

$$T_n^u = \frac{B_n}{r_n} u \tag{5}$$

The corresponding energy consumption of this process is:

$$E_n^u = p_n T_n^u = \frac{p_n B_n}{r_n} \tag{6}$$

The second part is the processing delay of MEC server processing computing tasks, which is defined as the number of computing resources allocated by MEC server for mobile user devices. The processing delay is expressed as:

$$T_n^c = \frac{D_n}{f_n} \tag{7}$$

During the period when the MEC server performs calculation, the mobile user device is in the state of waiting for receiving, and the idle power of the mobile user device in this state is, then the energy consumption during this period is:

$$E_n^c = P_n^w T_n^c = \frac{P_n^w D_n}{f_n} \tag{8}$$

For the last part, the return rate of wireless network is generally much higher than that of uploaded data, and the return execution result is much smaller than that of input data, so the execution delay and energy consumption are generally ignored. For the complete offloading calculation process, the execution delay and energy consumption are as follows:

$$T_n^e = T_n^u + T_n^c = \frac{B_n}{r_n} + \frac{D_n}{f_n} \tag{9}$$

$$E_n^e = E_n^{\mathrm{u}} + E_n^c = \frac{p_n B_n}{r_n} + \frac{P_n^w D_n}{f_n} \tag{10}$$

To sum up, the weighted total cost of execution delay and energy consumption for mobile user devices that choose offload computing is:

$$C_n^e = \theta_1 T_n^e + \theta_2 E_n^e \tag{11}$$

According to Equation (2)-(11), the sum of the total system weighted costs of all users can be obtained, that is, the system objective function is:

ISSN: 2688-8645

$$C_{all} = \sum_{n=1}^{N} (1-a_n)C_n^l + a_n C_n^e\} \tag{12}$$

4. Problem Model

Based on the above system model, computational offloading and resource allocation are expressed as optimization problems to minimize the total system cost. In order to minimize the total cost of the entire system, it is necessary to find the best uninstallation decision and computing resource allocation scheme that meets the requirements of computing tasks for users. The problem is described as follows:

$$\begin{aligned} &\min_{(A,f)} C_{all} = \min_{(A,f)} \sum_{n=1}^{N} (1-a_n)C_n^l + a_n C_n^e\} \\ &s.t.C1: a_n \in \{0,1\}, \forall_n \in N \\ &C2: T_n^l, T_n^e \le T_n^{\max}, \forall_n \in N \\ &C3: 0 \le f_n \le f_{\max}, \forall_n \in N \\ &C4: \sum_{n=1}^{N} a_n f_n \le f_{\max}, \forall_n \in N \end{aligned} \tag{13}$$

In the above formula, $A = \{a_1, a_2, a_3, ..., a_N\}$ is the offloading decision vector, and $f = \{f_1, f_2, f_3, ..., f_N\}$ is the resource allocation vector. C1 indicates that each mobile user device can only select local computing or uninstall computing to perform its computing tasks. C2 indicates that the delay caused by either local execution or uninstallation calculation cannot exceed the maximum allowable delay. C3 indicates that the MEC server cannot provide users with more computing resources than it can provide when the MEC server has limited resources. C4 means that the computing resources allocated to each mobile user device do not exceed the maximum computing resources that can be provided by the MEC server.

5. Problem Solutions

Reinforcement learning is used to solve the above problems. It is a kind of algorithm that allows the computer to operate completely randomly from the beginning, learn from mistakes through constant trial and error, and finally find the rules, so as to learn the method to achieve the goal. This is a complete reinforcement learning process. Let the computer in the continuous attempt to update their own behavior, so step by step to learn how to operate their own behavior to get high marks. It has three important elements: status, action, and rewards, and the goal is to get the most cumulative rewards.

5.1. Three Elements Definition

In this paper, the classical q-learning algorithm of reinforcement learning is used to transform the above problem of minimizing the weighted total cost into the problem of maximizing the reward. In the context of reinforcement learning in this paper, they are defined as follows:

To minimize the value of the objective function, the system state S consists of two parts S = (X, Y). Where, X represents the total cost of the system, and $X = C_{all}$ is the target to be optimized. The optimal state can be determined by observing the cost value of each state. Y is the number of idle resources on the MEC server, calculated by formula $Y = f_{\max} - \sum_{n=0}^{N} f_n$.

ISSN: 2688-8645

The actions of the system include the uninstallation decision vectors $A=\{a_1,a_2,a_3,...,a_N\}$ and computing resource allocation vectors $f=\{f_1,f_2,f_3,...,f_N\}$ of all users. Combining the two vectors, the action vectors can be obtained $a=[a_1,a_2,a_3,...,a_N,f_1,f_2,f_3,...,f_N]$, and the system actions determine the uninstallation decision and computing resource allocation scheme selected by users.

The reward function is set as $R(S,a)=\frac{X_{\text{local}}-X(S,a)}{X_{\text{local}}}$, which represents the reduction ratio of the total system cost in the current state compared with the total cost calculated locally. The larger $R(S,a)$ is, the smaller $X(S,a)$ is in the current state. The maximum reward means the minimum system weighted total cost, which successfully transforms the problem.

5.2. Q-learning Method

Q-learning is a method to record behavior value. Q is Q(S,a), which is the reward that can be obtained by taking action A at a certain moment of state S, and the environment will feedback corresponding reward R according to the agent's action. Therefore, the main idea of the algorithm is to build a Q-table of state and action to store Q value. Then use the Q value to select the action with the maximum reward. Its updating rule is, where S, A represents the current state and action, S',a' represents the next state and action of S. I is the learning rate, which is a constant satisfied. When I approaches 0, the agent mainly considers the immediate reward, while when I approaches 1, the agent also considers the future reward. Agents learn through experience and continue to explore from one state to another until they reach the target. Each exploration of agent is called an episode, and the value of Q(S,a) is continuously iterated until the optimal solution of the problem is obtained. The algorithm process is as follows:

Table 1. The algorithm process

Algorithm Q-learning method
1. Given the learning rate parameter I and the reward matrix 2.Initialize Q (S,a)=0 3.For each episode: 3.1 Randomly select an initial state S 3.2 If the target state is not reached, perform the following steps (1) Among all possible behaviors of the current state S, the maximum reward is selected a (2) Use the selected action A to get the next state S ' (3) Calculate Q(S,a) according to update rules (4) let s=s'

6. Simulation Results

6.1. Experimental Setup

In this paper, Python platform simulation is used to evaluate the performance of the proposed method, and experimental comparison and analysis are made with the whole local calculation method, the whole offloading calculation method and the random offloading calculation

ISSN: 2688-8645

method. Consider a single cell with 20 mobile user devices randomly distributed within 200 meters of the base station. The transmitted power and idle power of the mobile user device are set to p_n=100mv and $P_n^w = 10mv$ respectively. The bandwidth of the wireless channel is $W = 10MHz$. Suppose the data size of the upload task B_n is randomly distributed between (300,500), and the number of computing resources required D_n is randomly distributed between (900,1200).

6.2. Experimental Analysis

By setting different system parameters, the proposed Q-learning method is compared and analyzed with the whole local calculation method, the whole offloading calculation method and the random execution method. The whole local method means that all users choose to perform calculation on the local terminal. The full uninstall calculation method means that all users choose to uninstall the MEC server for calculation. The random execution method means that all users randomly choose to perform local computing or uninstall computing.

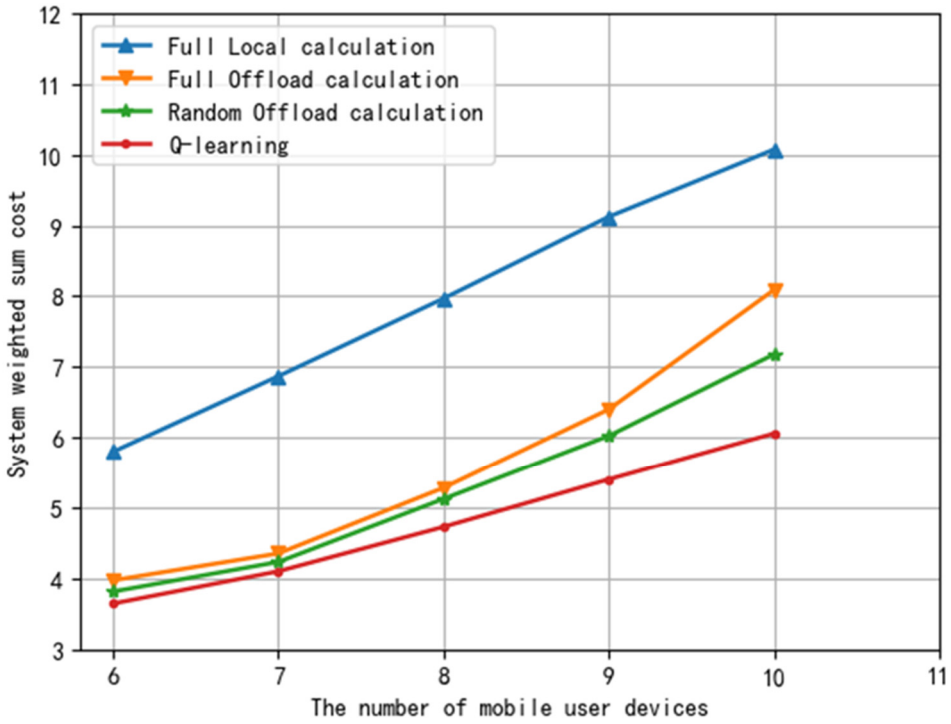

Figure 2. The relationship between the number of user devices and the weighted total cost

Figure 2 shows the relationship between the weighted total cost of the system and the number of mobile user devices, wherein the maximum computing capacity of MEC service is set as 5GHz/s. As can be seen from the changes of the four curves in the figure, with the increasing number of mobile user devices, the overall system-weighted total cost of the four methods presents an upward trend. In Figure 1, the method proposed in this paper has the best performance and can achieve good results. When the number of mobile user devices is less than 8, the difference between the weighted total cost of the system generated by random execution method and full offloading method and q-learning method is relatively small. However, with the increase of their number, the gap between them and Q-learning method is getting bigger and bigger, and the total offloading calculation method has the fastest growth. This is because the MEC server has limited computing resources, and the growing users have more fierce

ISSN: 2688-8645

competition for computing resources. As a result, the MEC server cannot provide sufficient computing resources to users, which increases the weighted total cost of the system. This group of experiments can draw a conclusion that we should make a reasonable offloading scheme for users, make full use of the computing resources of local terminals, reduce the competition among users, and rationally allocate the computing resources of MEC server, so as to minimize the total weighted cost of the system.

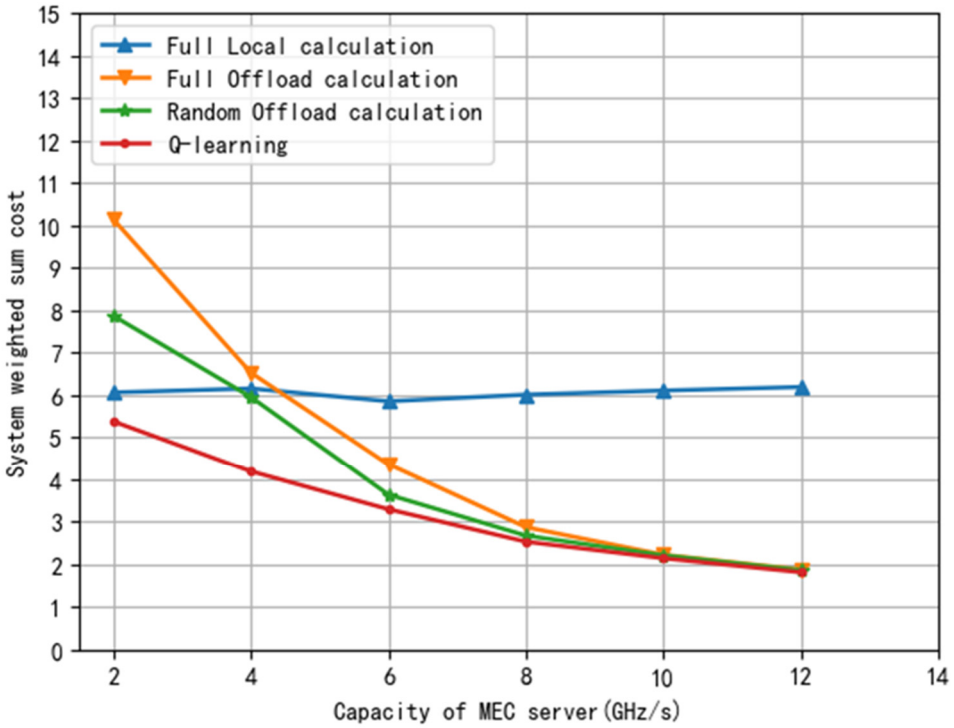

Figure 3. The relationship between MEC server computing power and total cost

Figure 2 shows the relationship between the weighted total cost of the system and the computing capacity of the MEC server, where the number of mobile user devices is set as 6. From the curve changes of the four methods in the figure, except for the whole local calculation method, there is almost no change, and the rest shows a significant downward trend. The local computing method remains unchanged because users only perform computing on the local terminal and do not need to use computing resources of the MEC server. The other three items decrease with the increase of the MEC server's computing power, because the larger the MEC server's computing power is, the sufficient computing resources can be allocated to users, thus reducing the processing time and the weighted total cost of the system. It can be seen from the figure that the overall effect of the method proposed in this paper is the best, and the curve is always at the bottom. When the computing power of MEC server is less than 8GHz/s, there is a large gap between the full offloading calculation method and the random execution method compared with q-learning method, but with the increasing of the computing power of MEC server, the gap between the three is almost no. Analysis shows that the user can not choose to offload all the computing resources in the case of limited computing resources, but in the case of computing resources overflow, relatively simple method can also achieve better results, so it is necessary to choose the best method according to the actual environment.

ISSN: 2688-8645

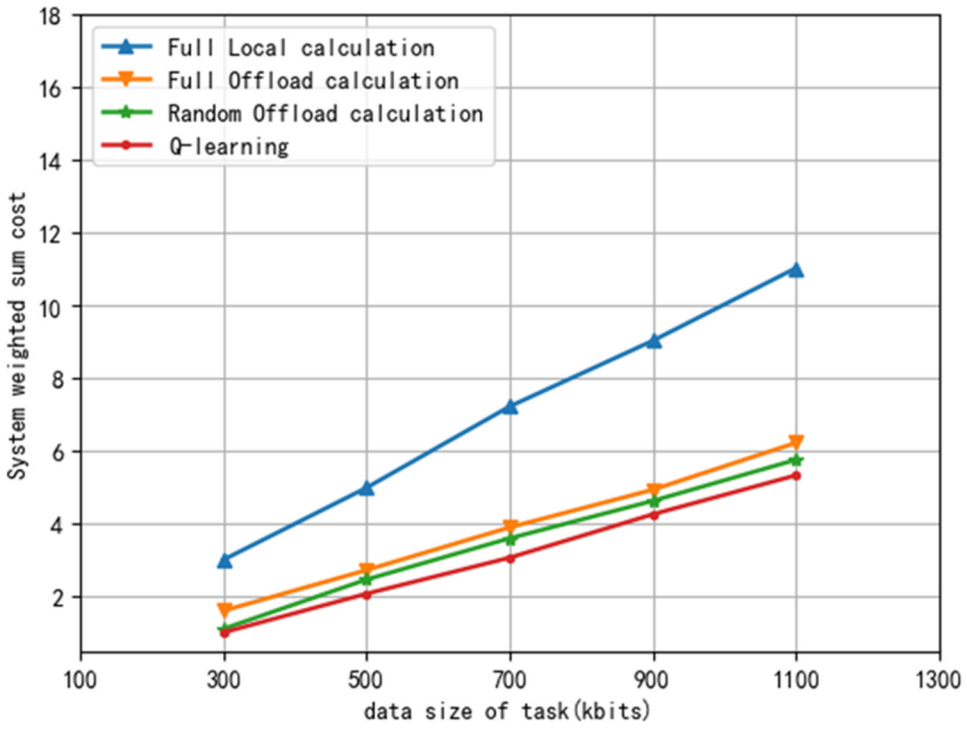

Figure 4. The relationship between the size of uploaded data and the total cost

Figure 3 shows the relationship between the weighted total cost of the system and the size of data uploaded by computing tasks. The number of mobile user devices is set as 5, and the maximum computing capacity of MEC server is 5GHz/s. From the curve changes of the four methods in the figure, the four methods all show an upward trend, which means that a larger amount of task data requires more time to upload and process, which will inevitably lead to an increase in delay and energy consumption in the transmission process and calculation process, thus causing an increase in the total weighted cost of the system. It can be seen from the figure that the method proposed in this paper can achieve the best effect, and its upward trend is relatively slow compared with other methods. The rising trend of the total system weighted cost produced by the whole local computing method is much higher than that of the other three methods. The analysis shows that when the amount of task data is too large, offloading computing can avoid the excessive use of local computing terminal resources, thus reducing the total system weighted cost.

7. Conclusion

In order to provide users with the best network service experience, this paper designs a computational offloading and resource allocation scheme based on reinforcement learning. Firstly, the moving edge computing communication model is established, and then the local computing and offloading computing models are listed. Finally, the weighted total cost of the system is optimized by integrating delay and energy consumption. Q-learning method is used to minimize the weighted total cost of the system. Experimental results show that, by setting different system parameters, the proposed method has the best overall performance compared with the other three benchmark methods, and can effectively reduce the total weighted cost of the system. In the future, with the increasing number of mobile user devices in the system, q-

ISSN: 2688-8645

learning method may have the problem of latitude disaster, so the next research direction is how to use more effective methods to solve the possible problems.

References

[1] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A Survey on Mobile Edge Computing: The Communication Perspective," in IEEE Communications Surveys & Tutorials, vol. 19, no. 4, pp. 2322-2358, Fourth quarter 2017, DOI: 10.1109/COMST.2017.2745201.

[2] K. Zhang, S. Leng, Y. He, S. Maharjan and Y. Zhang, "Mobile Edge Computing and Networking for Green and Low-Latency Internet of Things," in IEEE Communications Magazine, vol. 56, no. 5, pp. 39-45, May 2018, DOI: 10.1109/MCOM.2018.1700882.

[3] Tian Hui, Fan Shaoshuai, Lv Xinchen, Zhao Pengtao, He Shuo. Mobile edge calculation for 5G demand [J]. Journal of Beijing University of Posts and Telecommunications, 2017, 40 (02): 1 – 10.

[4] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang and W. Wang, "A Survey on Mobile Edge Networks: Convergence of Computing, Caching and Communications," in IEEE Access, vol. 5, pp. 6757-6779, 2017, DOI: 10.1109/ACCESS.2017.2685434.

[5] Liu J, Mao Y, Zhang J, et al. Delay-Optimal Computation Task Scheduling for Mobile-Edge Computing Systems[J]. IEEE, 2016.

[6] R Yu, Xue G, Zhang X. Application Provisioning in FOG Computing-enabled Internet-of-Things: A Network Perspective[C]//IEEE Infocom-ieee Conference on Computer Communications. IEEE, 2018:783-791.

[7] Chen X, Jiao L, Li W, et al. Efficient Multi-User Computation Offloading for Mobile-Edge Cloud Computing[J]. 2015.

[8] Kamoun M, Labidi W, Sarkiss M. Joint resource allocation and offloading strategies in cloud enabled cellular networks[C]// IEEE International Conference on Communications. IEEE, 2018.

[9] Zhang K, Mao Y, Leng S, et al. Energy-Efficient Offloading for Mobile Edge Computing in 5G Heterogeneous Networks[J]. IEEE Access, 2016, 4(99):1-1.

[10] Zhang J, Xia W, F Yan, et al. Joint Computation Offloading and Resource Allocation Optimization in Heterogeneous Networks With Mobile Edge Computing[J]. IEEE Access, 2018:19324-19337.

[11] Haitao ZHAO, Yinyang ZHU, Yi DING, Hongbo ZHU. Research on Content-aware Classification Offloading Algorithm Based on Mobile Edge Calculation in the Internet of Vehicles[J]. Journal of Electronics & Information Technology, 2020, 42(1): 20-27. doi: 10.11999/JEIT190594.

[12] Zhang Genshan, Liu Xu ning. Tasks split and offloding scheduling decision in mobile edge computing with limited resources[J].Computer Applications and Software, 2019, 36(10):268-273+278.