Research Article

Optimization of Mobile Edge Computing Offloading Model for Distributed Wireless Sensor Devices

Songyue Han ^{1,2} **Dawei Ma**¹ **Chao Kang** ³ **Wei Huang**¹ **Chaoying Lin**¹
and **Chunyuan Tian**¹

¹Communications Non-Commissioned Officer School, Army Engineering University of PLA, Chongqing 400035, China

²32705 Unit of PLA, Xi'an, Shaanxi 710086, China

³Xi'an University of Posts & Telecommunications, Xi'an, Shaanxi 710061, China

Correspondence should be addressed to Songyue Han; colincooper@sina.com and Chao Kang; kangchao@stu.xupt.edu.cn

Received 2 December 2021; Revised 17 January 2022; Accepted 26 January 2022; Published 28 February 2022

Academic Editor: Gengxin Sun

Copyright © 2022 Songyue Han et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The development and popularization of mobile Internet and wireless communication technology have spawned a large number of computation-intensive and delay-intensive applications. Limited computing resources and existing technologies cannot meet the performance requirements of new applications. Mobile edge computing technology can use wireless communication technology to offload data to be stored and computing tasks to the nearby assistant or edge server with idle resources. Based on the data offloading of distributed wireless sensor device to device communication, the architecture is designed and the basic framework of distributed mobile edge computing is constructed. To solve the problem of high mobile cloud computing technology, the offloading model of optimized mobile edge computing was proposed, and the stability and convergence of the proposed algorithm were proved. Finally, the system performance of the proposed algorithm is verified by simulation. The results show that the proposed algorithm can converge within a finite number of steps. Compared with other benchmark schemes, the proposed algorithm has better performance in reducing system energy consumption, reducing moving edge response delay and system total delay.

1. Introduction

With the growth of mobile web services and the growth of social networking applications, mobile data traffic is experiencing explosive growth. The increasing mobile traffic is mainly caused by emerging mobile device applications that require higher network throughput and more stringent network latency, something that current 4G wireless networks cannot achieve. 5G wireless networks will be standardized, increasing network capacity by a factor of 1,000 compared to 4G networks, and latency will be less than one millisecond. It now runs efficient and powerful applications with more computing power, storage, bandwidth, and power. Applications typically include computer vision image processing, optical character recognition, and augmented reality [1]. Mobile cloud computing is a collection of servers located in remote data centers that provide sufficient computing, storage, and network resources for mobile devices

[2]. MCC delays are caused by backhaul links, so long delays between users and the cloud become a challenge. In order to meet the network latency requirements of 5G wireless networks on MCC, a new network architecture is needed. Therefore, moving edge computing came into being [3]. Edge distributed devices use low-level signaling to share information. MEC discovers the location of devices by receiving information, provides network information and real-time network data service applications, and implements MEC through the model to benefit business and events [4]. The application estimates radio and network bandwidth congestion based on RAN real-time information, helping to make informed decisions to better serve customers. How to enhance the space of MEC cloud server and storage capacity has become the focus of research.

As wireless sensor networks begin to attract great interest of researchers, the footprint of wireless sensor networks can be seen in various fields [5]. The application provides

real-time information to estimate congestion in radio and network bandwidth, enabling informed decisions and better service for customers. Wireless sensor network is a technology generated through the miniaturization of radio components and sensor devices [6]. It is a wireless communication network composed of sensor nodes composed by some small devices with certain communication and sensing capabilities. Wireless sensor network (WSN) technology, as a more intelligent information technology after the Internet, has been concerned by many fields. Wireless sensor network technology is attracting more and more attention. MEC allows direct mobile communication between the core network and end users, while connecting users directly to the nearest cloud-enabled edge network [7]. Deploying MECs on base stations enhances computing power and avoids bottlenecks and system failures. How to optimize the offloading model of mobile edge computing by distributed wireless sensor devices has become a hot issue.

2. Related Work

Thanks to the continuous progress of information and communication technology, a large number of emerging intelligent Internet of Things applications have emerged, which require a large number of wireless devices to quickly perform low-latency and high-complexity computing tasks. Generally, wireless devices are small in size and have limited battery power supply, so the key challenge to be solved is how to increase the computing power of these devices and reduce computing latency [8]. At present, cloud computing can provide rich computing resources and powerful computing power, but the physical distance between cloud server and wireless terminal device is long, and the multihop routing and addressing transmission from the access network to the core network is required, which make cloud computing generally unable to meet the low latency requirements of some emerging applications run by wireless devices. For this reason, mobile edge computing technology came into being. In mobile edge computing, by configuring servers on the edge of wireless networks, computing resources are deployed on the side of wireless access networks to reduce the transmission time between wireless devices and computing servers and effectively meet the requirements of low-latency computing. It can be seen that mobile edge computing effectively integrates wireless communication network and mobile computing technology. Wang et al. realized PROFINET fieldbus communication based on edge devices and integrated the information collected by a large number of island devices together. Edge computing refers to the network edge perform calculations of a new type of calculation model, object at the edge of the computing operations including downstream data from cloud services, and uplink data from all Internet services [9]. Liu et al. carried out a scheme of computing offloading of multiple mobile devices and joint management of wireless network resources, but the main optimization objective of the literature was to minimize energy consumption without paying too much attention to system delay [10]. Chen et al. proposed a mobile device offloading algorithm, which can effectively reduce system delay of the algorithm by taking advantage of linear

characteristics limited by inequality in the optimization problem. A strong assumption is made in the algorithm, assuming that wireless network resources are sufficient. Moreover, the network resources allocated to each mobile device are in a fixed proportion to the computing tasks offloaded by the mobile device. However, in the actual mobile edge system environment, wireless network resources are limited, so the actual feasibility of this algorithm needs to be considered [11].

Wireless communication technology has also been rapidly developed, data acquisition system relying on wireless communication technology, and began to develop to wireless sensor network. The main working mode of wireless sensor network is to collect information through nodes and carry out communication and data transmission among nodes through wireless communication [12]. With the continuous research on wireless sensor network, its application is no longer limited to the military field and gradually extends from military weapons to antiterrorism and disaster relief, large-scale structural health monitoring, environmental monitoring, medical care and transportation support, and other fields. Moreover, the integrated circuit technology is becoming more and more mature, the reliability of hardware electronic components of various functional modules is becoming more and more high, and the reliability of wireless sensor network is becoming more and more stable. It can provide accurate information at different times, places, and environments, making wireless sensor networks gradually get more and more applications in people's lives [13]. Since MEC servers are not deployed on a large scale to cellular networks, most of the literature is theoretical. Due to the communication between mobile devices and MEC servers, computational offloading will incur extra costs in terms of delay and energy consumption [14]. Siavoshi and others proved the existence of game equilibrium, and put forward an effective balance algorithm, each mobile devices according to their own situation decision computing tasks uninstall strategy. The goal is to minimize their own application execution time delay. Total system delay is not taken into account [15]. Zeng et al. such as the main research Gui multiple mobile devices at the edge of the service node case computing tasks offload and resource allocation optimization problem put forward a kind of low time complexity of the algorithm for calculation of mobile equipment offloading and mobile edge server selection, and main optimization goal is to improve computing offloading efficiency and save mobile offloading at the edge of the cloud resources [16]. Li et al. proposed a mobile edge computing system architecture based on the central cloud, which further expanded the resources of mobile devices by utilizing the sufficient computing resources of the central cloud. This architecture is mainly applied to the mobile network, and the offloading strategy is designed according to the real-time situation of the central cloud to improve the practicability of the network. However, the limitation of wireless network resources and the allocation of network resources are not taken into account [17]. The allocation of wireless resources and computing resources is particularly important for MEC systems.

This paper uses wireless communication technology to unload the data to be stored and the tasks to be calculated to the edge server, designs the architecture of mobile edge computing, and constructs the basic framework of distributed mobile edge computing. Solve the problem of high delay caused by existing high mobile cloud computing technology.

3. Distributed Mobile Edge Computing Offloading Model

Mobile devices can use mobile edge computing technology to offload their computing tasks to THE MEC server, which performs computation-intensive or time-delay sensitive tasks instead of MD by collecting a large number of idle resources and storage space distributed at the edge of the network, thus, saving energy for the device.

3.1. MEC Architecture. To meet the ever-increasing device requirements, cloud services are being moved to the vicinity of mobile devices, the emerging edge computing paradigm considered in mobile networks. By moving computing tasks to edge servers rather than remote clouds, service response times can be significantly reduced, thereby improving the user experience. The traffic through the return link can also be alleviated [18]. The structure of the MEC is shown in Figure 1. Business processing time on the server is to compensate for long wireless transmission delays.

The architecture of mobile edge computing usually consists of user layer, edge computing layer, and cloud layer. The user layer is composed of mobile devices, and the edge computing layer is composed of mobile edge cloud servers located at the edge of the network. The cloud layer is mainly composed of cloud servers. Mobile devices at the user layer can make full use of computing, communication, and storage resources of mobile edge cloud through wireless access network. Mobile devices transmit their basic information to mobile edge cloud servers through wireless access networks. Edge of a mobile service node may be equipped with one or more edges in the cloud, compared with the computing and storage resources of mobile devices, mobile communications at the edge of the cloud server have richer, computing, and storage resources, can support mobile devices running time delay sensitive, large amount of calculation, or cache, etc, and also can carry out data real-time interaction [19]. The mobile edge cloud server in the edge computing layer receives real-time information and computing tasks unloaded by mobile devices. Mobile edge cloud servers are deployed on the edge of the network, and the distance between mobile devices is relatively short. At the same time, affected by physical scene factors, it has certain limitations compared with public cloud computing resources. Mobile edge cloud server can transfer part of computing tasks to the clouds, which can be executed by public cloud computing, which can realize clouds centralized management. Cloud refers to public cloud servers deployed in remote clouds. The mobile edge cloud server can send information to the cloud. The cloud can not only store long-term useful information but also carry out task processing and get

FIGURE 1: MEC architecture diagram.

the overall complete view of the covered area. However, offloading tasks from the mobile edge cloud server to the cloud also requires a certain transmission delay, so only nondelay-sensitive computing tasks can be offloaded. By providing global management and centralized control, the cloud's public cloud server provides a great help for the mobile edge cloud server to decide the optimal resource allocation strategy and the optimal computing offload strategy.

3.2. Distributed Mobile Edge Computing Framework. In distributed sensor network, each sensor can process its own information independently, provide a large amount of data, further obtain the classification characteristics of the target, and avoid the serious performance degradation of the single sensor system caused by electronic countermeasures. In the distributed fusion structure, each sensor can process its own information independently and then send each decision result to the data fusion centre for fusion. The basic architecture of edge computing is shown in Figure 2. Cloud servers are typically located in the core network, different from cloud computing, edge computing combines edge computing nodes into the network [20]. Edge computing can be run as a single computing platform or a collaboration platform with other components (including cloud).

To support real-time and interactive applications, mobile edge computing can store data on mobile devices on the edge, and the storage is distributed. The storage capacity of edge servers is still very limited compared to the resource-rich cloud. The storage types of data required by devices are extremely diverse. Therefore, edge servers need to have multiple types of storage policies to meet users' requirements. Different from the simple calculation provided by traditional caching and access technology, the calculation of edge server is more independent and tends to be intelligent [21]. Edge computing is closer to the terminal device, reducing the time delay and energy consumption of uploading computing tasks to the cloud, thus improving the quality of user experience. Mobile edge computing processes large amounts of raw data collected near different applications and performs real-time data analysis to generate valuable information. The ability to analyse data at the edge reduces the latency required to send data to the cloud and

FIGURE 2: Distributed mobile edge computing framework diagram.

wait for responses from the cloud. The results of local data analysis are then used to make decisions. Mobile edge computing helps entities make real-time decisions and actions based on well-processed data in an automated manner. Its decision-making ability improves system availability by reducing the exchange of components and data. Mobile edge computing enables remote control and monitoring, especially of critical equipment in insecure environments, including remote or more comfortable or secure locations. Mobile edge computing acts as an additional layer between the cloud and mobile devices to improve network security. Edge Cloud can be used as a secure distributed platform, providing security credential management, malware detection, software patch distribution, and trusted communications to detect, verify, and counter attacks. Because of the close proximity of mobile edge computing, it can quickly detect and isolate malicious entities and can initiate real-time responses to reduce the impact of attacks. This will help minimize service disruptions.

3.3. Offloading Model of Moving Edge Computing System. Assume that each user has a queue buffer that stores incoming but unprocessed computing tasks. In each time slot, the arrival process of user computing task is independent and identically distributed, and the average arrival rate is $\lambda I = E[A_i(t)]$. Meanwhile, each computing task can be processed locally or uninstalled to the MEC server [22]. Therefore, when the time gap t is fixed, the length vector of the household queue is

$$Q(t) = [Q_1(t), Q_2(t), \dots, Q_n(t)]. \quad (1)$$

The update process of $Q_i(t)$ is as follows:

$$Q_i(t) = \max \left\{ \frac{Q_i(t) + Y_i(t)}{D_{\Sigma,i}(t)} \right\}, \quad (2)$$

where the total amount of computing tasks processed by user I at time t is expressed as

$$D_{\Sigma,i}(t) = \frac{\tau f_i(t)}{L_i} + \frac{\sum_{j \in S} R_{ij}(t)}{L_i}. \quad (3)$$

The first part on the right of equation (3) is the amount of computing tasks processed locally by the user. $F_i(t)$ is the computing resources allocated by user i to process the computing tasks, that is, the CPU cycle frequency. L_i is the CPU cycle required to execute each bit of the computing task. The second part is the amount of computing tasks processed by offloading to MEC server. $R_{ij}(t)$ is the transmission rate when user i offloads computing tasks to MEC server J at time t , and its expression is

$$R_{ij}(t) = \zeta_{ij}(t) W \tau \log b + \frac{p_{ij}(t) h_{ij}(t)}{\zeta_{ij}(t) N W}, \quad (4)$$

where W is the bandwidth of MEC server, $\xi_{ij}(t)$ represents the proportion of bandwidth allocated by MEC server j to user i , $p_{ij}(t)$ and $h_{ij}(t)$, respectively, represent the transmission power and channel gain from user i to MEC server j , and N is the power spectral density of Gaussian white noise. In addition, since each base station is connected to a MEC server, j also refers to a MEC server in this article. Task request is dynamic, and the length of the task queue may exceed the user cache space, resulting in packet loss. Therefore, the task requirements of low delay and high reliability, a probability constraint, are added to the user queue length [20], namely,

$$\lim_{t \rightarrow \infty} p(Q_i(t) \geq Q_i^{\max}) \leq \varepsilon_i, \quad (5)$$

where $Q_i^{\max}$ stands for the queue threshold of user i , and ε_i stands for the overspill tolerance threshold of the task queue of user i , whose value is much less than 1. There are multiple queue buffers in each MEC server that can simultaneously store computing tasks offloaded by multiple users but not yet processed by the MEC server. Define the task queue of user i in MEC server j as $X_{ji}(t)$, and its update process is as follows:

$$X_{ji}(t+1) = \max \left\{ \frac{X_{ji}(t) - Y_{ji}(t)}{L_j} - \frac{\tau f_{ji}(t)}{L_i}, 0 \right\}, \quad (6)$$

$$Y_{ji}(t) = \min \left\{ Q_i(t) + Y_i(t) + \frac{\tau f_{ji}(t)}{L_i}, R_{ji}(t) \right\}, \quad (7)$$

where formula (7) represents the calculation that user i offloads to server j at time t , and $f_{ji}(t)$ represents the calculation that server j assigns to user i . Because MEC servers are deployed to provide users with more computing power, this article assigns the CPU cores of each server to at most one user to perform computing tasks. This paper also adds a probability constraint to the MEC server task queue length, namely,

$$\lim_{t \rightarrow \infty} p(X_{ji}(t) \geq X_{ji}^{\max}) \leq \varepsilon_i, \quad (8)$$

where $X_{ji}^{\max}$ represents the task queue threshold of user i in MEC server j , and ε_{ji} represents the task queue overflow tolerance threshold of user i in MEC server j , whose value is much less than 1.

3.4. Offloading Model Optimization of Moving Edge Computing System. According to the multitask distributed offloading method oriented to moving edge computing, its characteristics lie in that the uplink and downlink transmission rates in the steps are calculated by the following formula:

$$R_m^{y_1} = \log_2 \left(1 + P_{y_2} \eta \right), m = 1, 2, \dots, M. \quad (9)$$

Among them, the superscript $y_1 \in \{\text{UL}, \text{DL}\}$, respectively, ascending and descending link subscript m calculation the serial number of access points, the subscript $y_2 \in \{T, R\}$, respectively, transmitting and receiving mode, according to different mode of transmission rate of different transmission link, P_{y_2} said transmitting and receiving power, and m said system service count the number of access points of the current mobile station.

According to the moving edge computing-oriented multitask distributed offloading method described by rights, its characteristics lie in the mathematical optimization problems in the steps described are as follows:

$$T(x) = \max_{m \in i} \sum_{n \in j} x_{nm} \left\{ \frac{R_m^{\text{UL}}}{\alpha_n} + \frac{R_m^{\text{DL}}}{\beta_n} + \frac{r_m}{\gamma_n} \right\}, \sum_{n \in j} x_{nm} = 1, \quad (10)$$

where $i = \{1, 2, \dots, A\}$ and $j = \{1, 2, \dots, B\}$, respectively,

represent the set of tasks generated by mobile station and computing access points serving the mobile station, where elements A and B , respectively, represent the total number of tasks generated by mobile station and computing access points serving the mobile station; $X = [X_{nm}]B \times (A+1)$ is the task offloading access matrix; element X_{nm} represents the element in the N th row and the m column of matrix X , which represents the access parameter of the access point A when the access parameter task B is offloaded. $\psi(x) = \lambda t T(x) + \lambda e E(x)$, the elements λt and λe , respectively, represent the effects of delay and energy loss in the objective function in the current scenario, and the elements $T(x)$ and $E(x)$ represent delay and energy loss in the current scenario.

$$E(x) = p \sum_{n \in j} x_n \frac{\gamma_n}{r} + p \sum_{n \in j} x_{nm} \frac{\alpha_n}{R_m^{\text{UL}}} + p \sum_{n \in j} x_{nm} \frac{\beta_n}{R_m^{\text{DL}}}. \quad (11)$$

Elements α_n , β_n , and γ_n , respectively, represent the initial data size, the size of the task to be calculated, and the size of the output task after calculation. Element R_m represents the working rate of the computing access point M , and element P represents the power loss of the mobile station's local computing task, transmission task, and receiving task.

According to the said multitask distributed offloading method oriented to moving edge calculation, its characteristics lie in that the said method includes the following processes: reconstructing the matrix X , vectorizing the matrix X as x , where

$$x = [x_0, x_1, \dots, x_A] = [x_l]_{l \times (BA+B)}, \quad (12)$$

$$x_m = [x_{1m}, x_{2m}, \dots, x_{Bm}]. \quad (13)$$

Element $L = AB + B$ is the dimension of vector x , that is, the total number of elements of matrix x . The auxiliary variable $u = [x_l]_{l \times (AB+B)}$ is introduced, where u_l represents the probability of $x_l = 1$. It is transformed into a problem of finding the optimal probability u , constrained by $X_{Bm} \in \{0, 1\}$, and the probability density function of decision set x is defined as Bernoulli distribution:

$$p(x, u) = \prod_{i=1}^L u_i^{x_i} (1 + u_i)^{(1-x_i)}. \quad (14)$$

Transform the original equation into the minimum cross-entropy:

$$\min_{\pi} H = \max_{\pi} \frac{1}{S} \sum \ln p(x, u). \quad (15)$$

4. Simulation Results and Analysis

The effectiveness of the proposed offloading optimization algorithm for moving edge computing is verified by statistics and comparative analysis of simulation results. The edge computing system optimizes the computational unloading model of mobile devices and verifies whether the proposed

FIGURE 3: The influence of different mobile devices and different algorithms on total delay of the system.

optimization algorithm can reduce the total delay of system execution applications.

4.1. Model Validity Analysis. The analysis of the effectiveness of the offloading optimization algorithm for moving edge computing is mainly carried out through the following steps: in the case of different numbers of mobile devices, the influence of different algorithms on the average moving edge response delay. The influence of different algorithms on the total delay of the system is analyzed by line graph under different arrival rates of computational tasks. When the mobile device's own computing resources are different, the influence of different algorithms on the total delay of the system is analyzed by line chart.

The influence of different algorithms on the total system delay under different numbers of mobile devices is shown in Figure 3. In this simulation experiment, the range of numbers is [5,50], and the step size of the changing device is 4.

The experimental results show that the total system delay increases with the increase of the number of mobile devices. This is because the total amount of computing resources provided by wireless network resources and mobile edge cloud is fixed, and the competition for system resources increases when the number of mobile devices is large, so the total system delay also increases. The greedy algorithm only focuses on the shortest task execution delay of the device itself, and the total system delay is large. The competition for mobile edge cloud resources increases, and the total system delay required to perform all computing tasks increases. The total delay increases the fastest. In this algorithm, because all computing tasks are executed locally on mobile devices, computing resources of mobile edge cloud are not used, and mobile devices have limited resources. When a large number of computing tasks are executed, a large application execution delay will occur. For example, when the number of mobile devices is 50. In the mobile edge cloud first computing algorithm, all computing tasks generated by mobile devices are offloaded to the mobile edge cloud for execution. The execution delay of computing tasks is divided into two parts: wireless transmission delay and mobile edge cloud execution delay. Mobile edge cloud can expand computing resources for mobile devices, but the offloading of too many computing tasks will cause serious net-

FIGURE 4: Effect of different mobile devices on average moving edge response delay.

FIGURE 5: Effect of different mobile devices on average moving edge response delay.

work congestion for wireless network transmission, thus bringing large transmission delay. In addition, the computing resources of mobile devices themselves are also idle in the mobile edge cloud priority computing algorithm, resulting in a waste of resources. The optimization algorithm makes full use of the wireless sensor and mobile edge cloud computing in the system to make a decision on the offloading of mobile devices. Therefore, the total system delay required for application execution is less than the other three benchmark algorithms.

The influence of different numbers of mobile devices on average moving edge response time is shown in Figure 4. In this simulation experiment, the range of the number is set from [5,50], and the change step is 4.

The experimental results can be roughly observed in the moving edge computing system, the more mobile devices, the longer the average moving edge response time of the system. In the local first computing algorithm, all computing tasks of mobile devices are executed locally, so the moving edge response delay in the local first computing algorithm is always zero. Compared with greedy computing algorithm and mobile edge cloud first computing algorithm, the

FIGURE 6: Effect of transfer parameters on mobile computing offloading.

optimal computing offloading optimization algorithm proposed in this paper avoids the idle and waste of system computing resources, so the effect is better.

The impact of the average rate of task arrival on the total system delay is shown in Figure 5. The abscissa in the figure of experimental results is the average arrival rate of computing tasks for mobile devices. The range of average rate of computing tasks is controlled at $[0, 6]$, and the step size of change is 1.

It can be seen from the experimental figure that in the dynamic edge computing system, with the increase of computing requirements, the total system delay required to perform computing tasks also increases accordingly. The analysis results show that when the arrival rate of computing tasks increases, the total delay of the system does not increase significantly, and the average delay required for the execution of computing tasks decreases significantly. The optimization algorithm combining the optimal computing offloading and resource allocation can make full use of the wireless transmission and computing resources in the system, avoid the waste of resources, and reduce the total delay of the system.

4.2. Model Performance Analysis. The mobile edge computing system has a large number of mobile devices arriving every time, and the mobile devices are loaded with delay-sensitive applications, which will generate a large number of intensive computing tasks according to the use requirements of mobile devices. The transmission resources in the mobile edge computing system are not fixed, so the optimal computing unloading optimization algorithm needs to maintain good stability to ensure that the decision results of the algorithm have less impact on the whole system when the resources in the system change. This paper mainly focuses on the influence of the change of transmission parameter C on mobile device computing task offloading and system delay. The influence of the change of transmission parameter C on mobile device computing offloading is shown in Figure 6. The abscissa in the figure of experimental results is the transmission parameters of wireless network transmission resources, that is, the amount of data actually

needed to transmit a single computing task, and the ordinate in the figure is the sum of all computing tasks offloaded by mobile devices.

The number of mobile devices in this simulation experiment is set to 50. As can be seen from the experimental results, with the forcing-port of transmission parameter C , the offloading amount of total computing tasks of mobile devices shows a downward trend. This is because the transmission parameter C represents the transmission cost. The larger c is, the larger the actual data amount required by the calculation task of the transmission unit is. In this case, the congestion of the network will be aggravated and the transmission delay of the system will be longer. The experimental results show that when the transmission parameter C is less than 3.0, there is no significant impact on the computing offloading strategy of mobile devices. When the transmission parameter C is greater than 3.0, mobile devices in the system are more inclined to perform computing tasks locally, because offloading computing tasks at this time will bring large transmission delay.

Analysis results can be found that the transmission parameter c smaller, less than 2, had no significant effect on the system time delay, prove that the change of the transmission parameter c in the edge of mobile computing system calculation uninstall not significantly affect the decision-making organ, and wireless network at this time the actual transmission data volume is low, so the transmission lower than the proportion of the total delay system. With the increase of transmission parameter C , especially when parameter C is greater than 3.0, the actual transmitted data volume of wireless network increases significantly. At this time, mobile devices will have a large transmission delay in offloading computing tasks. Therefore, the optimal computing offloading optimization algorithm is more inclined to leave computing tasks in the local execution of mobile devices.

4.3. Influence of Model Iteration Times on Total Energy Consumption. The performance of this algorithm is verified in offloading model optimization, and different strategies of offloading optimization algorithm based on moving edge

FIGURE 7: The influence of iteration times on total energy consumption of the system.

computing are compared. Simulation experiments were carried out in Matlab software, and simulation scenes were built based on the described multiuser system model. Figure 7 shows the influence of the number of iterations of simulation experiment on the total energy consumption of the system.

The convergence performance of different algorithms is compared in the figure. DGWO1 algorithm and DGWO2 algorithm converge gradually to the local optimal solution after the 50th and 60th iterations, respectively, and the convergence trend is slow. This is because the operation of crossover function may lead to the loss of the optimal individual in the next generation population, and the phenomenon of losing the optimal individual may occur repeatedly in the whole position update process. The algorithm in this paper increases the amount of information contained in each individual by expanding dimensions, so the accuracy is higher. Moreover, the algorithm in this paper combines cosine convergence factor, which can make the algorithm better jump out of local optimum.

5. Conclusion

In mobile edge computing technology, the time delay required to perform computing tasks is very important. When mobile devices uninstall computing tasks, it may occur those multiple mobile devices uninstall computing tasks through the same wireless access point, and the off-loaded computing tasks are executed in the same mobile edge cloud service section. By optimizing the offloading model of mobile edge computing, the wireless transmission parameters in the system are well supported, and the impact on the whole system is small. The performance meets the system requirements of mobile edge computing. The optimal computing offloading optimization algorithm can make full use of mobile device's own computing resources, wireless network transmission resources, and mobile edge cloud computing resources in the mobile edge computing system to avoid waste and idle resources. At the same time, compared with the local computing first algorithm and the mobile edge cloud computing first algorithm, the joint optimization algorithm of optimal computing offload and

resource allocation can better reduce the mobile edge response delay and the total delay of the system. Computing, storage, network, and communication resources are deployed at the edge of the mobile network, reducing network operations and service delivery delays, and improving user experience. In addition, MEC reduces the transmission bandwidth requirements for the core network by deploying servers at the edge of the network, reducing operating costs. In the next step, the combination of deep reinforcement learning and computational unloading is considered to design a more intelligent unloading algorithm to adapt to the complex and changeable edge unloading environment.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by Army Engineering University of PLA.

References

- [1] X. Cao, G. Zhu, J. Xu, and K. Huang, "Optimized power control for over-the-air computation in fading channels," *IEEE Transactions on Wireless Communications*, vol. 19, no. 11, pp. 7498–7513, 2020.
- [2] W. Z. Khan, E. Ahmed, S. Hakak, I. Yaqoob, and A. Ahmed, "Edge computing: A survey," *Future Generation Computer Systems*, vol. 97, pp. 219–235, 2019.
- [3] Z. Ziming, L. Fang, and C. Zhiping, "Edge computing: platforms, applications and challenges," *Journal of Computer Research And Development*, vol. 55, no. 2, p. 327, 2018.
- [4] Y. He and Z. Tang, "Strategy for task offloading of multi-user and multi-server based on cost optimization in mobile edge

- computing environment,” *Journal of Information Processing Systems*, vol. 17, no. 3, pp. 615–629, 2021.
- [5] G. Li, Q. Lin, J. Wu, Y. Zhang, and J. Yan, “Dynamic computation offloading based on graph partitioning in mobile edge computing,” *IEEE Access*, vol. 7, pp. 185131–185139, 2019.
 - [6] Z. Lv, D. Chen, and Q. Wang, “Diversified technologies in internet of vehicles under intelligent edge computing,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 4, pp. 2048–2059, 2020.
 - [7] D.-R. Chen, L.-C. Chen, M.-Y. Chen, and M. Y. Hsu, “A coverage-aware and energy-efficient protocol for the distributed wireless sensor networks,” *Computer Communications*, vol. 137, pp. 15–31, 2019.
 - [8] Q. Liu, Z. Chen, J. Wu, Deng, Liu, and Wang, “An efficient task scheduling strategy utilizing mobile edge computing in autonomous driving environment,” *Electronics*, vol. 8, no. 11, p. 1221, 2019.
 - [9] R. Wang, M. Li, L. Peng, Y. Hu, M. M. Hassan, and A. Alelaiwi, “Cognitive multi-agent empowering mobile edge computing for resource caching and collaboration,” *Future Generation Computer Systems*, vol. 102, pp. 66–74, 2020.
 - [10] M. Liu, K. Yang, and N. Zhao, “Intelligent signal classification in industrial distributed wireless sensor networks based industrial internet of things,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 7, pp. 4946–4956, 2021.
 - [11] C. Chen, L. Chen, L. Liu et al., “Delay-optimized v2v-based computation offloading in urban vehicular edge computing and networks,” *IEEE Access*, vol. 8, pp. 18863–18873, 2020.
 - [12] Y. Sun, C. Song, S. Yu, Y. Liu, H. Pan, and P. Zeng, “Energy-efficient task offloading based on differential evolution in edge computing system with energy harvesting,” *IEEE Access*, vol. 9, pp. 16383–16391, 2021.
 - [13] H. Sun, M. Liu, Z. Qing, X. Li, and Li, “Energy consumption optimisation based on mobile edge computing in power grid internet of things nodes,” *International Journal of Web and Grid Services*, vol. 16, no. 3, pp. 238–253, 2020.
 - [14] A. H. Sodhro, S. Pirbhulal, and V. H. C. De Albuquerque, “Artificial intelligence-driven mechanism for edge computing-based industrial applications,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 4235–4243, 2019.
 - [15] S. Siavoshi, Y.-S. Kaviani, M. Tarhani, and H. F. Rashvand, “Geographical multi-layered energy-efficient clustering scheme for ad hoc distributed wireless sensor networks,” *IET Wireless Sensor Systems*, vol. 6, no. 1, pp. 1–9, 2016.
 - [16] F. Zeng, Y. Chen, L. Yao, and J. Wu, “A novel reputation incentive mechanism and game theory analysis for service caching in software-defined vehicle edge computing,” *Peer-to-Peer Networking and Applications*, vol. 14, no. 2, pp. 467–481, 2021.
 - [17] H. Li, K. Ota, and M. Dong, “Learning IoT in edge: deep learning for the Internet of Things with edge computing,” *IEEE Network*, vol. 32, no. 1, pp. 96–101, 2018.
 - [18] C. Wang, Y. Lv, Q. Wang, D. Yang, and G. Zhou, “Service-oriented real-time smart job shop symmetric CPS based on edge computing,” *Symmetry*, vol. 13, no. 10, p. 1839, 2021.
 - [19] L. Lombardo, S. Corbellini, and M. Parvis, “Wireless sensor network for distributed environmental monitoring,” *IEEE Transactions on Instrumentation and Measurement*, vol. 67, no. 5, pp. 1214–1222, 2018.
 - [20] M. Chen, Y. Miao, and H. Gharavi, “Intelligent traffic adaptive resource allocation for edge computing-based 5G networks,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 2, pp. 499–508, 2020.
 - [21] A. Chatterjee, P. Venkateswaran, and D. Mukherjee, “A unified approach of simultaneous state estimation and anomalous node detection in distributed wireless sensor networks,” *International Journal of Communication Systems*, vol. 30, no. 9, p. e3191, 2017.
 - [22] Z. Lv, D. Chen, R. Lou, and Q. Wang, “Intelligent edge computing based on machine learning for smart city,” *Future Generation Computer Systems*, vol. 115, pp. 90–99, 2021.