Journal of Computers Vol. 32 No. 6, 2021, pp. 15-23
doi:10.53106/199115992021123206002

A Security Edge Computing Offloading Solution for 5G Cellular Network

Yue Ma[1], Xin Li[2], Jianbin Li[3*]

[1] Assistant General Managers, State Grid Jibei Information Telecommunication Company, Beijing, China
mayue@sina.com

[2] Deputy Director of Data, State Grid Jibei Information Telecommunication Company, Beijing, China
sherry_lixin@aliyun.com

[3] School of Control and Computer Engineering, North China Electric Power University, Beijing, China
lijb87@ncepu.edu.cn

Received 10 November 2020; Revised 4 March 2021; Accepted 15 March 2021

Abstract. As 5G communication technology becomes more mature and more widely used, countless infrastructures will be connected to the network. Shifting computing tasks from user equipment (UE) to the edge of the network, the need for mobile edge computing (MEC) to break hardware and resource constraints is increasing. However, when many users offload tasks to the edge server, the network load and transmission interference will increase, so this paper proposes a security edge computing offload model for 5G cellular networks. In this model, we novel consider safety and cost costs. In addition to these factors, energy consumption and delay are the main considerations for this model. To verify the validity of our model, a simulation experiment was designed. Simulation results show that this model can achieve multi-objective optimization of energy consumption, delay, security, and cost in a 5G environment, significantly reduce energy consumption, and achieve resource optimization decisions.

Keywords: 5G cellular network, edge computing, computing offload, multi-objective optimization

1 Introduction

In December 2018, the Central Economic work Conference defined for the first time the orientation of "new infrastructure construction" in 5G, artificial intelligence, industrial Internet, Internet of things and other areas [1]. In 2019 and 2020, several policies were successively introduced, emphasizing the acceleration of the construction of "new infrastructure". Mobile cloud computing is the main mode of service delivery and application in the Internet of things. Mobile cloud computing allows mobile devices to partially or completely migrate local computing tasks to cloud servers, which solves the problem of resource shortage of mobile devices and saves energy consumption for local task execution [2]. However, unloading the tasks to the cloud server located in the core network needs to consume the resources of the return link, resulting in additional delay overhead, which cannot meet the requirements of low delay and high reliability in big data scenarios. In 2014, the European Telecommunications Standards Association creatively proposed mobile edge computing. The system allows devices to upload computing tasks to network edge nodes, such as base stations and wireless access points, which not only meets the need of expanding computing capability of terminal devices, but also makes up for the disadvantage of long-time delay in cloud computing [1]. As a major change point in the field of mobile communication, 5G is the leading field of "new infrastructure". The combination of MEC and 5G technology not only enables the Internet of things, ultra-high-definition video and car networking, but also integrates with cloud computing, big data and artificial intelligence.

* Corresponding Author

Considering the limited battery capacity and computing power of mobile terminals, it is very difficult to deal with computing-intensive tasks in mobile terminals [2]. Therefore, it is necessary to unload the computing task to the edge computing server, and the unloading of computing task is one of the key issues in the field of mobile edge computing.

The uploading of computing in the mobile edge computing network needs to follow the uploading strategy [1]. According to the performance requirements of unloading, the common unloading strategies are divided into three main types: minimizing delay, minimizing energy consumption and maximizing benefits. In terms of minimizing delay unloading decision methods: Literature [1, 2] introduces the online learning allocation strategy based on dynamic computing unloading for all computing unloading of a single mobile terminal. A time interval is set to detect the time delay and the failure rate of uninstall during each time interval to decide whether to uninstall. In [3-5], the problem of task unloading is expressed as a mixed integer nonlinear computation process, a complete polynomial time approximation scheme, and a game theory approach to design the distributed computing unloading algorithm, to achieve lower computational time overhead. All the above strategies have achieved the goal of reducing the execution time delay in the process of unloading, but the energy consumption of mobile terminal is not considered, if the mobile terminal battery is exhausted quickly, the corresponding unloading strategy cannot be used normally.

In terms of minimizing energy consumption unloading decision methods: In [6] an off-line precomputation unloading strategy is proposed, which transforms the problem of minimizing energy consumption under time delay constraints into a constrained Markov decision process problem. Literature [7] uses a greedy algorithm to solve the problem of minimizing energy consumption and proposes a priority algorithm for the largest energy saving task. Literature [8-9] proposes a binary particle swarm optimization algorithm, which uploads the computational portion of a single mobile terminal device and reduces the energy consumption of the user's mobile terminal by 25%. In [10], an entry unloading algorithm for optimal resource allocation based on TDMA system is proposed, and the problem of resource allocation in mobile edge computing offload system is discussed. The strategy in [10] is further improved by [11], which proposes an optimal resource allocation access unloading algorithm based on TDMA and OFDMA systems, in which radio and computing resources are decomposed into finer granularity indices for optimal allocation. The strategy of minimizing energy consumption is to seek the algorithm of minimizing energy consumption under the limitation of mobile terminal's delay. Such a strategy depends on the transmitting power of the mobile terminal and the quality of the radio channel. Most of the existing strategies are verified in the form of simulation and draw conclusions, but the real unloading conditions may not be restored in the process of simulation. It may also ignore the time-varying quality of the radio channel and the mutual interference of multiple mobile terminals when calculating and unloading at the same time. In addition, some unloading strategies remain at the level of theoretical guidance and have not been realized technically.

In terms of maximizing benefits unloading decision methods: Literature [12] proposed a game strategy for computing uploading of multiple mobile terminals. In [13-14] presents a complete unload strategy for heuristic based on semidefinite relaxation and randomization mapping. Literature [15] proposed a Lyapunov optimization calculation unloading strategy based on dynamic computational unloading. Literature [16-17] minimizes system overhead by minimizing communication and computing resource overload. Literature [18] solves the optimization problem of resource allocation by using the mutual iteration of the game algorithm and the Hungarian algorithm, thereby reducing the calculation energy consumption and delay, and minimizing the total system overhead. In essence, the calculation unloading strategy of maximizing benefits is to analyze the influence of time delay and energy consumption on the calculation of total unloading consumption by analyzing the influence of time delay and energy consumption in the process of performing calculation unloading under the condition of meeting the execution time limit and energy consumption limit. to find a balance point to make the time delay or energy consumption limit setting more suitable for the actual scene, so as to achieve the goal of minimizing the total cost, that is, to maximize revenue.

Through the analysis and comparison of the above three kinds of computing unloading decisions, we can see that compared with the calculation on the mobile side, the most significant advantage of uploading computing to the edge server is that it can reduce the computing delay. The energy consumption can be further discussed under the condition that the calculation delay is guaranteed. Although the strategy of maximizing revenue cannot minimize delay or energy consumption, it can be closer to specific application requirements. However, the security and cost factors are ignored in the above three kinds of computing unloading decisions, so this paper proposes a mobile edge computing unloading method based on edge computing offloading model to carry out effective computing unloading and resource allocation. The model can achieve multi-objective optimization of energy consumption, time delay, safety and cost in 5G environment, and significantly reduce energy consumption, so that the delay and energy consumption of task execution can be balanced.

The structure of this article is as follows. Chapter 2 proposes the edge computing offloading model. Chapter 3 introduces experimental simulation. Chapter 4 summarizes the work of this paper.

2 Models

To enhance the end-user experience, meet the growing demand of high reliability and low delay computing services, and reduce the network load pressure, this chapter will analyze the status of user equipment, relay and edge server in the network. On the premise of fully considering the service capability of user equipment and edge server, we construct an edge computing upload model for 5G cellular network, to achieve the best resource scheduling scheme.

2.1 System Model

The system model is shown in Fig. 1. The roles included are user, relay, and edge server, and the three constitute a network. The following is an analysis of the status and functions of the three.

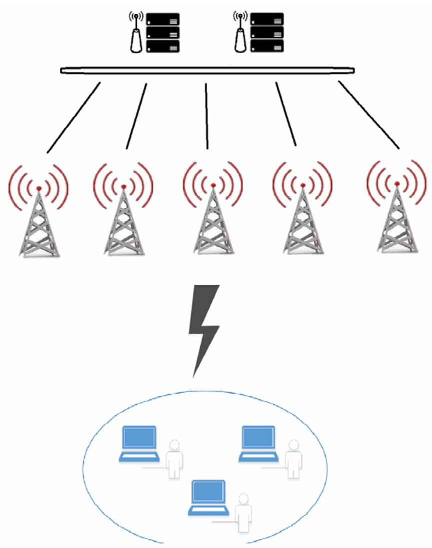


Fig. 1. 5G cellular network scenario with integrated edge computing upload

USERS: Users enjoy the services of relay and edge servers. Each user can define a different amount of computing tasks and choose a service scheme to compute locally or upload to an edge server.

RELAY: Relay provides users with data transmission and unload decision services. Assume that the computing task requested by the user is b_i. In order to provide better service and lower total cost, relay combines delay, energy consumption and other factors to make strategy decision and put forward appropriate optimization scheme.

EDGE SERVER: The edge server provides computing services to users. When the user chooses to unload the task to the edge server-side computing scheme, the data is transferred to the edge side through relay. The edge server sets the available computing resources in the unit of CPU cycle per unit time. Then it calculates the tasks accordingly and relays the final results back to the user.

In this paper, a multi-device scenario is considered, that is, K users are deployed together with M relays and N edge servers. If each user terminal has one computing task to complete, which can be performed locally or unloaded to the edge server. Here, we only consider two-state unloading of computing tasks, that is, tasks, intelligently choose to unload all to the edge server or execute all locally.

The two-state unloading strategy includes two kinds: unloading strategy and no-unloading strategy. Our goal is to minimize the overall cost of system users during task upload. Since both task transmission and calculation consume energy, we jointly consider these two aspects when modeling the problem. In addition, in order to ensure the security and reliability of the data, we add the key algorithm into the model and quantify it in the form of encryption and decryption delay. Next, we introduce the computing task processing model of this scheme in detail.

2.2 Computing Task Processing Model

Because the task can be performed on the edge server or on the user's device. Therefore, the computing task processing model is divided into two parts: unloading computing model and local computing model. The offload computing model represents the use of computing resources of the edge server to process tasks after the relay decides to unload to the edge server. The local computing model represents the use of computing resources of the user equipment to perform tasks.

A. Unloading calculation model

The whole model consists of the following steps: firstly, the user encrypts the task data and then uploads it to the relay through 5G network, and then the relay forwards to the edge server. After that, the edge server allocates corresponding computing resources to decrypt and process the task. Finally, the calculation results are returned to the client according to the original path.

Because the communication of this model relies on 5G network, the wireless connection between entities is mainly composed of forward link and return link. Transmission delay is an important index that affects the final choice. Therefore, the parameters related to data transmission between users, relay and edge server are defined.

For the front-end link connecting users and relays, the achievable data rate is calculated as follows:

$$r_{i,m} = Wlog_2[1+(p_{i,m}\times g_{i,m})/(s^2+\sum_{l=1}^{M}\sum_{j=1,j}^{K} p_{j,l}\times g_{j,m})] \quad \textbf{(1)}$$

Among them, user $i\in(1,K)$, relay $m\in(1,M)$, W is channel bandwidth, $p_{i,m}$ is transmission power on forward link from user to relay, $g_{i,m}$ is channel gain between user and relay, σ^2 is Gaussian noise.

Similarly, the achievable data transfer rate on the backhaul link from the relay to the edge server is expressed as:

$$\mathrm{r}_{m,n} = W\log_2[1+(p_{m,n}\cdot g_{m,n})/(\sigma^2+\sum_{q=1}^{N}\sum_{l=1,l\neq m}^{M} p_{l,q}g_{l,n})] \quad \textbf{(2)}$$

Among them, the edge servers $n\in(1,N)$, $p_{m,n}$ are the transmission power on the backhaul link relayed to the edge server, $g_{m,n}$ represents the channel gain between the two.

Because the user may upload data to the edge server at any time after sending the task information to the relay. Therefore, the total delay experienced by the computing task includes encryption delay, decryption delay, waiting delay, etc., which can be calculated by the following formula:

$$\Delta_{\mathrm{i}} = \Delta_{i,m}^{t} + Q^{i} + \Delta_{m,n}^{t} + \Delta_{n} + \Delta_{\mathrm{encryption}} + \Delta_{\mathrm{decryption}} \quad \textbf{(3)}$$

Let Δ_i the total delay of task computing experience. Where, represents the transmission delay when the user sends the task to the relay, b_i represents the number of tasks uploaded by the user. $Q^i = b_i/\lambda$ represents the waiting time of tasks uploaded to the relay waiting for the edge server to allocate resources, λ represents the average rate in the backhaul link. $\Delta_{m,n}^{t} = \sum_{i=1}^{K} {}_m b_i / r_{m,n}\cdot\Delta_n$ represents the transmission delay when a task is sent from the relay to the edge server. $\Delta_n = (\sum_{i=1}^{K} {}_m\omega_i)/f_o$ represents the computing delay of tasks on the edge server, ω_i represents the resource consumption of each user task on the mobile edge server, f_o is the available computing resources of the edge server, which is provided in CPU cycles per unit time. $\Delta_{\mathrm{encryption}} = b_i\cdot\Delta_{encrytion}^{t}$ represents the total encryption time of the task, $\Delta_{\mathrm{encryption}}^{\mathrm{t}}$ is the encryption time per unit data volume. $\Delta_{\mathrm{decryption}} = b_i\cdot\Delta_{decrytion}^{t}$ is the total decryption time of the task,

$\Delta^{t}_{decryption}$ is the decryption time of unit data volume. ω_i represents the resource consumption of each user task on the mobile edge server.

The energy consumption of unloading scheme is expressed as follows:

$$e_i^{MEC} = p_{i,m} \Delta_{i,m}^t + p_{(\quad)} \times \Delta \quad + P \quad (Q^i + \Delta_n + \Delta_{m,n}^t + \Delta \quad _n) \tag{4}$$

e_i^{MEC} represents the total energy consumption of the client under the offload policy. $p_{i,m}\Delta_{i,m}^t$ represents the energy cost of the user to upload the task to the relay, $p_{i(encryption)}$ and $P_{standby}$ represent encryption power and waiting power respectively.

Cost consumption is considered in the unloading strategy, which is defined as cost = data volume * unit flow unit price (unit flow unit price c_{up}) and its calculation formula is as follows:

$$C = b_i \bullet c_{up} \tag{5}$$

In order to minimize the total cost of system users, we put forward the following optimization scheme by integrating various factors (W1, W2 and W3 are the weights of corresponding characteristic factors, and the sum is 1):

$$J_{MEC} = min \sum_{i=1}^{K} (w1 \quad \Delta_i + w2 \quad e_i^{MEC} + w3 \quad C \tag{6}$$

B. Cost calculation model

When the relay decides that the task is executed locally by the user, the local computing rate is r_i, the local computing time is $\Omega_i = b_i / r_i$, and the client's computing power is p_i, then the energy consumption is expressed as:

$$e_i^{LOC} = p_i \cdot \Omega_i \tag{7}$$

To minimize the total cost of the client, the calculation method of the optimization scheme is as follows:

$$J_{LOC} = min \sum_{i=1}^{K} (v1 \quad \Omega_i + v2 \quad e_i^{LOC} + v3 \quad C) \tag{8}$$

Since the local computing strategy is selected, 5G traffic is not used, so the cost function $C = 0$.

3 Experiment

To verify that the proposed method can be effectively applied in the 5G, the following simulation experiments are done. The simulation experiment of this paper uses Python platform to train the genetic algorithm model. The results show that the unloading decision can reach the optimum. The results show that the results can be distributed in a wider range.

3.1 Setup

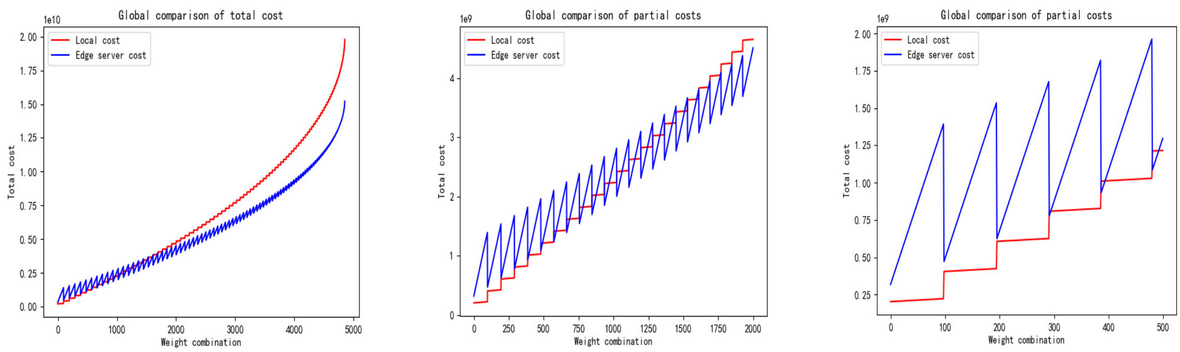
Suppose that there are *n* edge nodes, one base station and one edge computing server in the wireless network scenario of a single plant area. The topology is as shown in the figure. In this experiment, we assume that each task has only two kinds of decision-making. The calculation tasks are performed locally and unloaded to the edge calculation server, which are represented by 0 and 1 respectively. Each node has limited computing power but can compute its own tasks. The edge computing server can perform computing tasks, and it is assumed that its performance can be as large as possible, which is larger than the calculation amount of any node but smaller than the total calculation capacity of all nodes. Base station adopts centralized edge computing unloading strategy, which plays a scheduling role in edge computing. This paper assumes that the computing power is large enough, the scheduling queue is small, and the scheduling time can be ignored. As a decision-maker, the base station decides the calculation execution right of the node. When calculating the time delay, because the return time delay is very small, it can be ignored. The parameters set in the experiment are shown in Table 1.

Table 1. Parameter setting

Parameter	Range of value	Parameter	Range of value
n	(5, 100)	$t_n^{off} - a_n$	(0, 10s)
Vn	(300, 500)	f_n	5GHz/s
S_n	(500, 1000)	P_n	100MW
a_n	(0, 60s)	P_n'	100MW
B	20MHz	g_n	10dB
n_0	-88dB	$I_n(f_n')^2$	1MJ/kg(L)

3.2 Results

First, we consider the impact of the three weights w_1, w_2, w_3 on the results. Fig. 2(a) is the comparison of the total cost of the edge server and the local side corresponding to the combination of different weights when the weight granularity is two decimal places. The overall trend in the figure changes with the increase of the energy consumption weight w_2, corresponding to each w_2. The value of in each small period in the figure changes with the change of w_1 and w_3. Fig. 2(b) shows the first 2000 values in Fig. 2(a). It can be seen from the figure that when w_2 is less than 0.04, the cost of the edge server is higher, and when w_2 is greater than 0.21, the cost of local calculation is higher. Fig. 2(c) shows the first 500 values in Fig. 2(a).

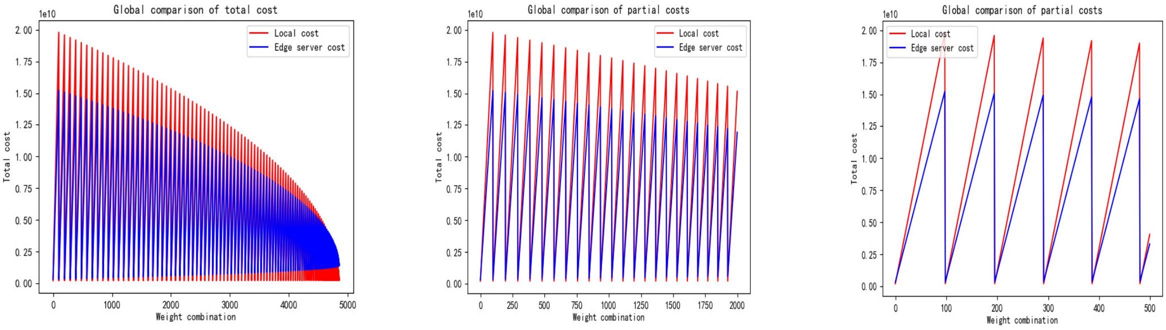


(a) (b) (c)

Fig. 2. the comparison of the total cost with w_2 increases

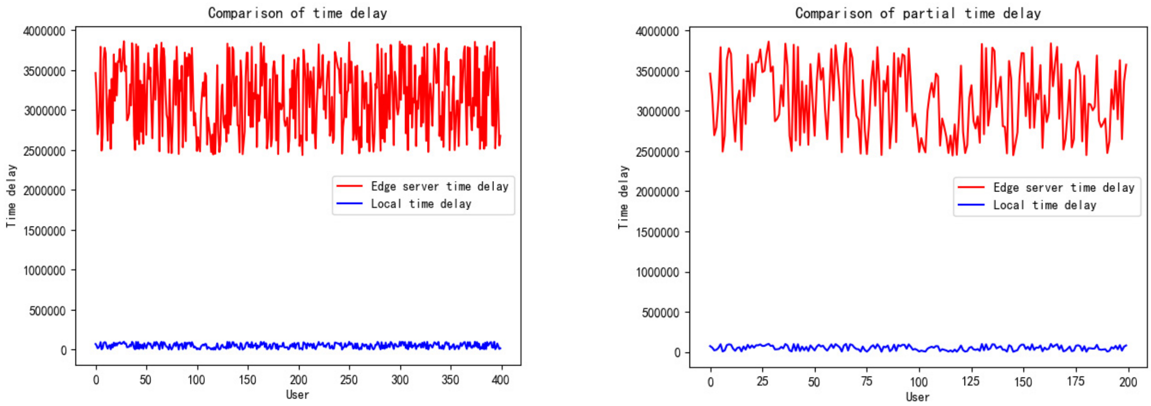
Fig. 3(a) is a comparison of the total cost of the edge server and the local side corresponding to the combination of different weights when the weight granularity is two decimal places. The overall trend in the figure changes as the energy consumption weight w_1 increases, corresponding to each w_1. The value of in each small period in the figure changes with the change of w_1 and w_3. Fig. 3(b) shows the first 2000 values in Fig. 3(a). Fig. 3(c) shows the first 500 values in Fig. 3(a).

Next, we consider the impact of energy consumption, delay, and cellular network data volume cost on the solution of the multi-objective optimization problem. Fig. 4(a) shows the comparison of the delay between the edge server and the local. Fig. 4(b) shows the first 200 values in Fig. 4(a). Fig. 5(a) is a comparison of energy consumption between the edge server and the local. Fig. 5(b) shows the first 200 values in Fig. 5(a). Fig. 6(a) is a comparison of cellular network data volume cost between the edge server and the local. Fig. 6(b) shows the first 200 values in Fig. 6(a).



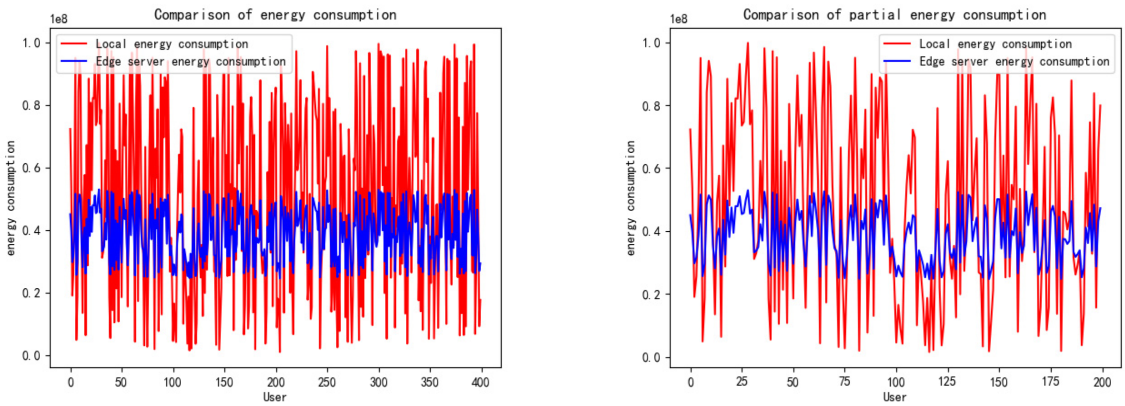
(a) (b) (c)

Fig. 3. The comparison of the total cost with w_1 increases



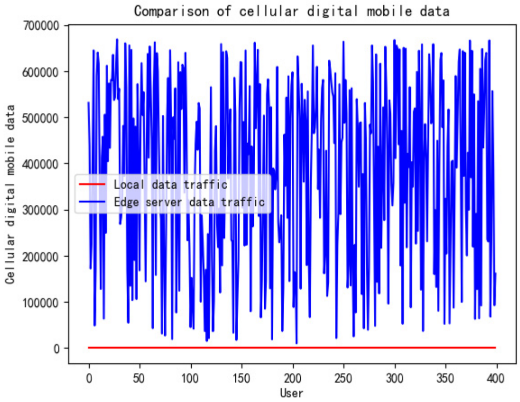
(a) (b)

Fig. 4. The comparison of the delay between the edge server and the local

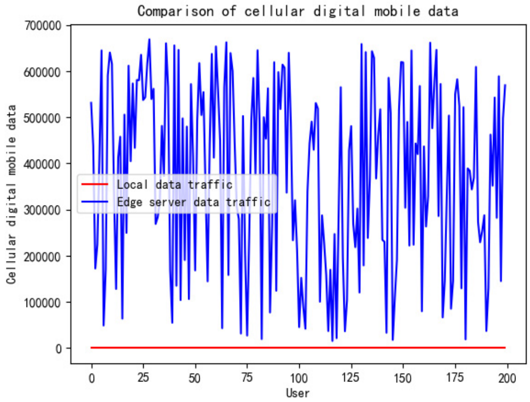


(a) (b)

Fig. 5. The comparison of the energy consumption between the edge server and the local



(a)



(b)

Fig. 6. The comparison of cellular network data volume cost between the edge server and the local

4 Conclusion

This paper proposes an edge computing offload model for 5G cellular networks. In this model, we novel consider safety and cost costs. In addition to these factors, energy consumption and delay are the main considerations for this model. To verify the validity of our model, a simulation experiment was designed. Simulation results show that this model can achieve multi-objective optimization of energy consumption, delay, security, and cost in a 5G environment, significantly reduce energy consumption, and achieve resource optimization decisions.

But there are still some problems in this paper. First, the multi-objective optimization problem of genetic algorithm is essentially a search problem without self-learning ability, which is also needed in the target optimization problem of computational offload. Finally, the setting of the experimental scene is relatively simple, without considering the more complex scene.

Acknowledgements

This work is supported by the China Science and technology project Nos. “5700-201914241A-0-0-00” and “Z181100005118016”.

References

[1] M. Yuyi, J. Zhang, K.-B. Letaief, Dynamic computation offloading for mobile-edge computing with energy harvesting devices, IEEE Journal on Selected Areas in Communications 34(12)(2016) 3590-3605.

[2] S. Ulukus, A. Yener, E. Erkip, O. Simeone, M. Zorzi, P. Grover, K. Huang, Energy harvesting wireless communications: A review of recent advances, IEEE Journal on Selected Areas in Communications 33(3)(2015) 360-381.

[3] M. Chen, Y. Hao, Task offloading for mobile edge computing in software defined ultra-dense network, IEEE Journal on Selected Areas in Communications 36(3)(2018) 587-597.

[4] R. Yu, G. Xue, X. Zhang, Application provisioning in fog computing-enabled internet-of-things: A network perspective, in: Proceedings of IEEE INFOCOM 2018-IEEE Conference on Computer Communications, 2018.

[5] X. Chen, L. Jiao, W. Li, X. Fu, Efficient multi-user computation offloading for mobile-edge cloud computing, IEEE/ACM Transactions on Networking 24(5)(2015) 2795-2808.

[6] K. Mohamed, W. Labidi, M. Sarkiss, Joint resource allocation and offloading strategies in cloud enabled cellular networks, in: Proceedings of 2015 IEEE International Conference on Communications (ICC), 2015.

[7] W. Feng, S. Chen, W. Zou, A greedy algorithm for task offloading in mobile edge computing system, China Communications 15(11)(2018) 149-157.

[8] D. Maofei, H. Tian, B. Fan, Fine-granularity based application offloading policy in cloud-enhanced small cell networks, in: Proceedings of 2016 IEEE International Conference on Communications Workshops (ICC), 2016.

[9] K. James, R.-C. Eberhart, A discrete binary version of the particle swarm algorithm, in: Proceedings of 1997 IEEE International conference on systems, man, and cybernetics. Computational cybernetics and simulation, 1997.

[10] C. You, K. Huang, Multiuser resource allocation for mobile-edge computation offloading, in: Proceedings of 2016 IEEE Global Communications Conference (GLOBECOM), 2016.

[11] C. You, K. Huang, H.-H. Chae, B.-H. Kim, Energy-efficient resource allocation for mobile-edge computation offloading, IEEE Transactions on Wireless Communications 16(3)(2016) 1397-1411.

[12] X. Chen, L. Jiao, X. Fu, Efficient multi-user computation offloading for mobile-edge cloud computing, IEEE/ACM Transactions on Networking 24(5)(2015) 2795-2808.

[13] M.-H. Chen, B. Liang, M. Dong, A semidefinite relaxation approach to mobile cloud offloading with computing access point, in: Proceedings of 2015 IEEE 16th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), 2015.

[14] M.-H. Chen, M. Dong, B. Liang, Joint offloading decision and resource allocation for mobile cloud with computing access point, in: Proceedings of 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2016.

[15] M. Yuyi, J. Zhao, S.-H. Song, K.-B. Letaief, Power-delay tradeoff in multi-user mobile-edge computing systems, in: Proceedings of 2016 IEEE Global Communications Conference (GLOBECOM), 2016.

[16] T. Zhao, S. Zhou, X.-Y. Zhao, Z. Niu, A cooperative scheduling scheme of local cloud and internet cloud for delay-aware mobile cloud computing, in: Proceedings of 2015 IEEE Globecom Workshops (GC Wkshps), 2015.

[17] X. Guo, R. Singh, T. Zhao, Z. Niu, An index based task assignment policy for achieving optimal power-delay tradeoff in edge cloud systems, in: Proceedings of 2016 IEEE International Conference on Communications (ICC), 2016.

[18] J. Zhang, W. Xia, F. Yan, L. Shen, Joint computation offloading and resource allocation optimization in heterogeneous networks with mobile edge computing, IEEE Access 6(1)(2018) 19324-19337.