

Research Article

Dynamic Game-Based Computation Offloading and Resource Allocation in LEO-Multiaccess Edge Computing

Haoyu Wang,¹ Hengli Wang,² and Jianwei An ¹

¹School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China

²School of Computer Science (National Pilot Software Engineering School), Beijing University of Posts and Telecommunications, Beijing 100876, China

Correspondence should be addressed to Jianwei An; anjianwei@ustb.edu.cn

Received 29 April 2021; Accepted 14 July 2021; Published 8 November 2021

Academic Editor: Long Zhang

Copyright © 2021 Haoyu Wang 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 offloading of computing tasks in edge computing has always been a research hotspot and difficulty in recent years. As an effective way to run various applications on mobile devices with limited resources, it has been extensively studied by scholars from all walks of life. However, the traditional ground-based network-based edge computing network architecture cannot meet the needs of edge users with limited geographic areas. Therefore, this paper proposes an LEO (low earth orbit) satellite-based multiaccess edge computing network architecture and establishes a differential game model for this architecture. To obtain the Nash equilibrium solution of the open loop and the Nash equilibrium solution of the feedback for the task offloading amount, the relationship between the user's income and the QoE level under the optimal task offloading amount is finally analyzed and discussed.

1. Introduction

Mobile communication technology and wearable device technology have of late years grown with leaps and bounds, more and more tasks and data are generated at the edge of the network “far” from the cloud center, and the powerful computing power of the cloud center cannot be effectively displayed. In addition, mobile devices are gradually entering the stage of history, and various mobile wearable devices are playing an increasingly important role in people's daily lives [1]. This also makes cloud computing unable to meet the needs of such large data and dense geographical distribution and the time delay requirement of the task. Edge computing came into being to complement the shortcomings of cloud computing, and the computing offloading in edge computing is the main way to solve the above problems. However, due to the geographical constraints (ocean and mountain area) of users and the variety of tasks, the traditional edge computing network architecture based on the ground-based network cannot serve users well.

At present, some scholars have carried out some research work on the integration of space and edge computing tech-

nology. But in these works, the satellite network is still regarded as a relay network, ignoring the deployment of an edge server directly in a satellite network, so as to offload computing tasks to a space-based network or air-based network [2]. Directly processing the request tasks of edge users on satellite can not only reduce the traffic burden of the return link but also effectively reduce the delay of task processing, to improve the QoE (quality of user experience). However, the satellite resources are limited, and the on-board processing of the edge computing model is still faced with challenges such as computing overhead, maintenance cost, and intersatellite transmission. In this paper, a multiaccess edge computing offloading architecture based on LEO satellite is proposed, which fully considers the limited factors faced by the satellite in a specific scenario with limited resources, aimed at solving the problem of how to compute unloading and resource allocation when the user is limited in the field area.

2. Related Works

Since the mobile terminal is light and portable, its size and weight are restricted. As a result, the CPU processing

capacity of mobile terminals, the connection capacity of wireless and mobile networks, the storage capacity of hardware devices, and battery capacity and life are bound to be limited [3]. This determines that mobile terminals are resource-constrained. So as to solve the problem of resource constraints of mobile terminals and the decline of QoE levels due to resource constraints, computing offloading and migration provide an effective and direct way to solve the above problems [4].

In the field of computational offloading, in recent years, scholars mainly study partial offloading [5–11] and complete offloading [12–16] of computational offloading.

Zhang et al. [17] used the method of replication dynamics to obtain the evolutionary equilibrium of the evolutionary game, thereby realizing the optimal allocation of limited bandwidth resources and limited computing resources of edge service providers. By studying the problem of multiuser computing offloading in a multiwireless edge computing environment, Chen et al. [18] modeled the decision-making process of distributed multiuser computing offloading as a multiuser computing offloading game and designed an effective distributed computing offloading algorithm to solve the model. Chen et al. [19] solved the user offloading decision and resource allocation optimization problem of nonconvex quadratic-constrained quadratic programming and proposed an effective approximate solution. Pham et al. [20] study the problem of computing offloading in heterogeneous networks with multiple edge computing servers. Under the premise of minimizing the system overhead, users can choose one of the multiple servers to perform the task of computing and unloading in the system architecture they propose. Zhang et al. [5] design an efficient and energy-saving calculation and offloading scheme considering the multiple-access performance of 5g isomerism networks. Based on a greedy heuristic method with the linear Gini coefficient, Zhao et al. [6] constructed a joint optimal offload decision model to allocate wireless resources and computing resources on the premise of minimizing the energy expenditure of the multiuser edge computing system. Mao et al. [7] proposed a dynamic computing migration algorithm based on the Lyapunov optimization (LODCO) to optimize the execution delay of applications. The algorithm uses energy-harvesting technology to minimize the energy loss in the local execution phase and uses the power control method to optimize the energy cost for data transmission. This method can also prevent packet loss in the process of computing migration.

Ahn et al. [8] designed an energy resource-faced task scheduling scheme that can reduce energy expenditure and improve computational performance, which maximizes the gain of all users. Tao et al. [9] proposed a user energy expenditure minimization problem and used the Lagrange multiplier method to get the optimal offloading strategy, which can consider the user's quality of service and energy expenditure in a balanced way. Considering the correlation between intensive deployment of edge computing servers and user computing tasks, Dai et al. [10] proposed a two-tier computing offload framework. Through this framework, they can design an effective computing offload scheme to

solve the problem of load balancing between multiedge computing servers and achieve the purpose of minimizing system energy expenditure. Deng et al. [11] assumed that the application running on the device is composed of several parts with dependency and defined the computational migration problem as a 0-1 model. Finally, a binary particle swarm optimization (BPSO) algorithm was used to solve the problem.

All the above researches are based on the calculation of the ground-based network. However, in practical application, because of the complexity of topography and the diversity of special task requirements, the traditional foundation edge computing network cannot meet this challenge, so many scholars began to explore the obstacles caused by the combination of satellite communication and edge computing to customer service geography.

Wang et al. [12] proposed a satellite ground fusion network with dual edge computing capability to reduce the energy consumption and delay of the network. Zhang et al. [13] designed a collaborative computing offload method in the satellite edge computing scenario and used Network Functional Virtualization (NFV) technology to integrate the computing resources within the coverage of low earth orbit (LEO) satellites. Qiu et al. [14] constructed a satellite ground network framework based on a Software-Defined Network (SDN) to dynamically manage and coordinate the network, cache, and computing resources. Alsharoa and Alouini [15] proposed an efficient optimization framework to provide offloading services for ground-based network users by ground-based base stations, HAPs, and satellites. Xu et al. [16] constructed an air-ground-sea-integrated network architecture with edge and cloud computing functions, so as to provide flexible hybrid computing capacity for maritime services.

3. Motivation and Contribution

As mentioned above, thanks to the development of Internet technology, wireless communication technology, and the popularity of wearable intelligent devices, more and more practical application scenarios need the service mode of edge computing. However, the existing research shows that when the requirements of task quantity, delay, and energy consumption sensitivity are not enough for network edge processing, it is necessary to offload part or all of the tasks to the edge server (or cloud server) with stronger performance, so as to obtain better service performance. However, for some types of tasks, they can be completed at the edge device or on the edge server. For such tasks, how to make the unloading decision to achieve the optimal benefits, quality of service, and user experience? Therefore, the purpose of this paper is to study how to make the offloading decision between the edge device and the edge server for the continuous and divisible task types to achieve the highest QoE of user experience, as well as the internal relationship between the offloading decision component and device performance and network status. At the same time, the geographic area of the target edge server that uses LEO satellites to service

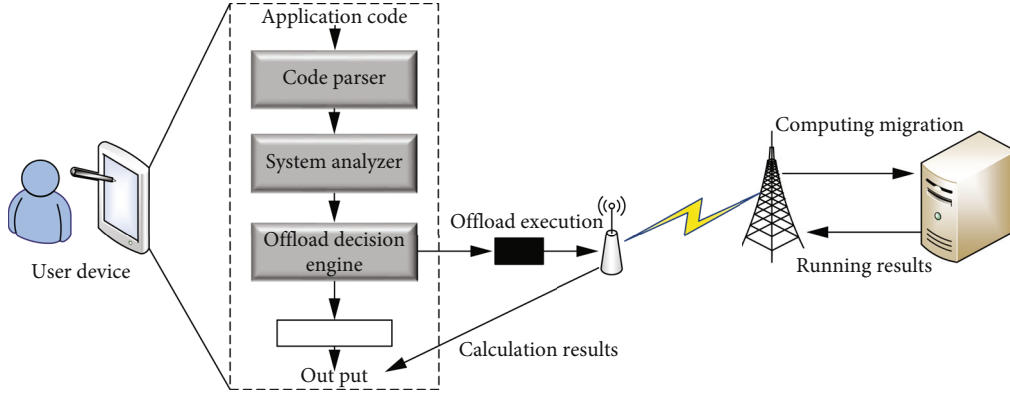


FIGURE 1: General form of computing offload in edge computing environment.

offloads is complex and rugged. The main contributions of this paper are as follows:

- (1) By using the LEO satellite to construct the LEO-MEC multiaccess converged network architecture, the problem of poor service performance of traditional edge server base stations and other facilities is solved when the geographical area of edge users is rugged and complex terrain
- (2) A game model of task offloading for edge users is established by using the differential game with the benefit of edge users as the goal and the QoE level as the constraint, and the optimal amount of task offloading for users with the Nash equilibrium solution of the open loop and the Nash equilibrium solution of the feedback is obtained

The rest of this paper is organized as follows: Section 2 introduces the related work and research in the field of edge computing and satellite fusion network. Section 4 describes the establishment process of the system model. In Section 5, the differential game is used to solve the model, and the open-loop and feedback Nash equilibrium solutions are obtained. In Section 6, the optimal strategy is simulated and verified. The experimental results show that the proposed method has a certain practical significance. Finally, the thesis summarizes the whole text and discusses the future work.

4. System Model

Edge computing, as the complement of cloud computing, is mainly to “sink” part of the cloud center applications to the edge server, so as to solve the needs of edge user intensive tasks and also to some extent reduce the burden of the core network. Computing offload is the main method to resolve this problem. It is a general form of edge computing task offloading, as shown in Figure 1 [21].

As shown in Figure 2, we consider establishing LEO-MEC multiaccess converged network architecture with edge servers deployed on LEO satellites (this paper assumes that the orbital altitude of the satellite is between 500 km and 1200 km). The

reason why LEO satellite is used is that most of the tasks processed by edge users are time delay-sensitive tasks (virtual reality, intelligent medical treatment, intelligent transportation, etc.), and the computing tasks are offloaded on the equipment with limited resources and these intensive tasks are offloaded to the edge server with stronger computing power. We consider the many-to-one scenario where multiple users offload an edge server. If the geographical area of the edge user is flat and wide (that is, the wireless transmission is not greatly affected by geographical factors), we will offload the computing task to the edge server in the nearby ground-based network; if the geographical area of the edge user is a rugged and complex mountainous area (that is, the wireless signal transmission is greatly affected by geographical factors), we will offload the computing task to the edge server in the corresponding space-based network.

We assume that the tasks in the sequence are discontinuous and divisible; that is, these tasks can be processed either on the client or on the edge server, or they can be unloaded to the edge server. This is what we usually call three modes: local mode, offload mode, and part-offload mode. In this paper, we mainly consider the mode of computing offload. In the architecture mentioned above, edge users can get benefits by completing a certain amount of tasks. When their performance is not enough to support the local execution of the task, they need to offload all or part of the calculation, which requires certain resources and revenue consumption. Therefore, how to reasonably plan the amount of offloading tasks is the purpose of this paper. Considering that the task offloading is performed, the consumption of task waiting time caused by transmission delay is inevitable. Therefore, we need to introduce a definition called the time penalty coefficient before building the model.

Definition 1. Time penalty coefficient $\mathbf{r}_i(\mathbf{t})$:

$$r_i(t) = \begin{cases} \phi, & 0 \leq t \leq 1, \\ \phi \ln t, & 1 < t < \infty, \\ C_{ap}\phi, & t \rightarrow \infty. \end{cases} \quad (1)$$

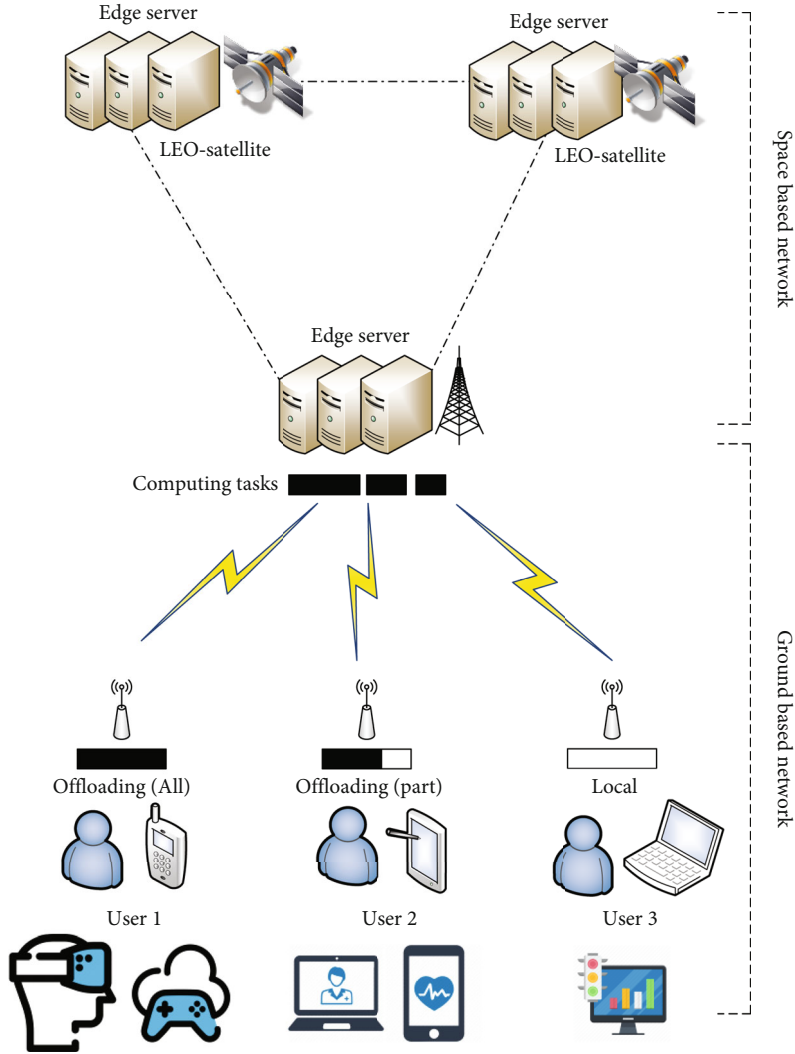


FIGURE 2: Mobile edge computing architecture based on low earth orbit satellite.

Because the benefits are timely, that is, at a certain QoE level, the faster the task is completed, the higher the expected earnings will be. Therefore, a penalty item is set here, which increases slowly with time until it reaches the upper penalty limit $C_{ap}\varphi$, which is similar to the actual set of package capping. The main consideration here is that when the task waiting time is too long, the system will drop the task and proceed to the next task. And that will reduce the overall benefits of edge users [22].

We use $U_i(t)$ to indicate the task offloading amount of user i at t time (this paper assumes that the marginal users are individually rational, that is, to maximize their own interests without considering the income of others) and $y_i (y_i \in (0, 1))$ to represent the proportion of offloaded tasks to the total task quantity, so we get the total task quantity $U_i(t)$:

$$U_i(t) = \frac{u_i(t)}{y_i}. \quad (2)$$

We use $x(t)$ to represent the QoE level of the LEO-MEC system. According to [23, 24], the mapping method between

QoE and QoS is shown in Figure 3. Since this paper is about computing offload and most of the tasks of edge users are delay-sensitive, we select the indicators related to transmission delay in Figure 3, so we can get

$$dx(t) = \left[\lambda x(t) + \varphi \times \sum_i u_i(t) \right] dt, \quad (3)$$

where $\lambda, \lambda \in (0, 1)$ is the weighted coefficient of the QoE level and $\varphi, \varphi \in [-1, 1]$ is the QoE index factor based on task completion quality. Note that in this paper, we mainly consider the increase or decrease in the QoE level due to the calculation offloading, so in the above formula, we use the task offloading amount $u_i(t)$ instead of the overall task amount $U_i(t)$.

We use R to represent the task revenue per unit, and $\chi u_i(t)x(t)$ is used to indicate the increase or decrease in income affected by QoE due to calculation and offloading, so we get the revenue function $G_i(t)$:

$$G_i(t) = RU_i(t) - \chi u_i(t)x(t) - r(t)u_i(t)\xi, \quad (4)$$

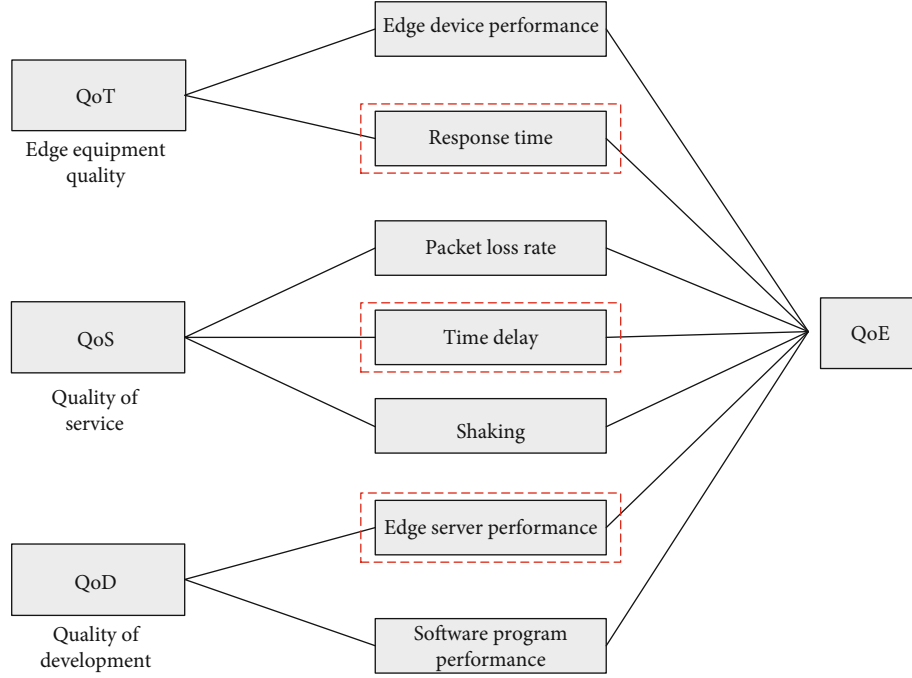


FIGURE 3: The mapping of QoS and QoE.

where ξ is the transmission delay with the offloading amount of $u_i(t)$. We use V' to represent the CPU efficiency of the edge server, W to represent the transmission bandwidth, p to represent the transmission power of the edge device, and N to represent the Gaussian white noise in the channel, so we can get

$$\xi = \frac{u_i(t)}{V'} + \frac{u_i(t)}{W \log_2(1 + (p/N))}. \quad (5)$$

As we consider that users are individually rational, in our many-to-one LEO-MEC framework scenario, when multiple users offload at the same time, they will compete with each other, resulting in the decline of their revenue and QoE level, as shown in Figure 4.

At this point, ξ becomes

$$\xi = \frac{\sum_{k=1}^{k=i} u_k(t)}{V'} + \frac{u_i(t)}{\left(u_i(t)/\sum_{k=1}^{k=i} u_k(t)\right) W \times \log_2(1 + (p/N))}. \quad (6)$$

So we get the income function $G_i(t)$:

$$G_i(t) = R \frac{u_i(t)}{y_i} - \chi u_i(t) x(t) - r(t) u_i(t) \cdot \left[\frac{\sum_{k=1}^{k=i} u_k(t)}{V'} + \frac{u_i(t)}{\left(u_i(t)/\sum_{k=1}^{k=i} u_k(t)\right) W \times \log_2(1 + (p/N))} \right]. \quad (7)$$

Given the time interval $[0, T]$, the edge node hopes to obtain the optimal amount of task offloading $u_i(t)$, so as to

maximize the benefits obtained by completing the task, as follows:

$$\begin{aligned} \max_{u_i(t)} G_i(t) = & \int_t^T \left\{ R \frac{u_i(t)}{y_i} - \chi u_i(t) x(t) - r(t) u_i(t) \right. \\ & \cdot \left. \left[\frac{\sum_{k=1}^{k=i} u_k(t)}{V'} + \frac{u_i(t)}{\left(u_i(t)/\sum_{k=1}^{k=i} u_k(t)\right) W \times \log_2(1 + (p/N))} \right] dt \right\}, \\ \text{s.t. } dx(t) = & \left[\lambda x(t) + \varphi \times \sum_i u_i(t) \right] dt, k \leq Q_E, \end{aligned} \quad (8)$$

where Q_E is the maximum capacity of the edge server to process user tasks. We will use the idea of the differential game to obtain the Nash equilibrium solution of the open loop and the Nash equilibrium solution of the feedback for the above model in the next section.

5. Game Analysis

5.1. Open-Loop Nash Solution of the Game. To maximize the revenue obtained in the process of completing tasks, the edge device should control the number of offloaded tasks $u_i(t)$ to get the Nash equilibrium solution of the open loop. To obtain the Nash equilibrium solution of the open loop, $u_i^*(t)$ is optimal if the inequality in formula (9) holds for all control strategies $u_i(t) \neq u_i^*(t)$ in the feasible region.

$$G_i(u_i(t), x(t), t) \leq G_i^*(u_i^*(t), x^*(t), t). \quad (9)$$

The Stackelberg equilibrium solution of the open loop for the problem consists of a group of control $u_i^*(t)$, in which $x^*(t)$ is the corresponding state trajectory. If the

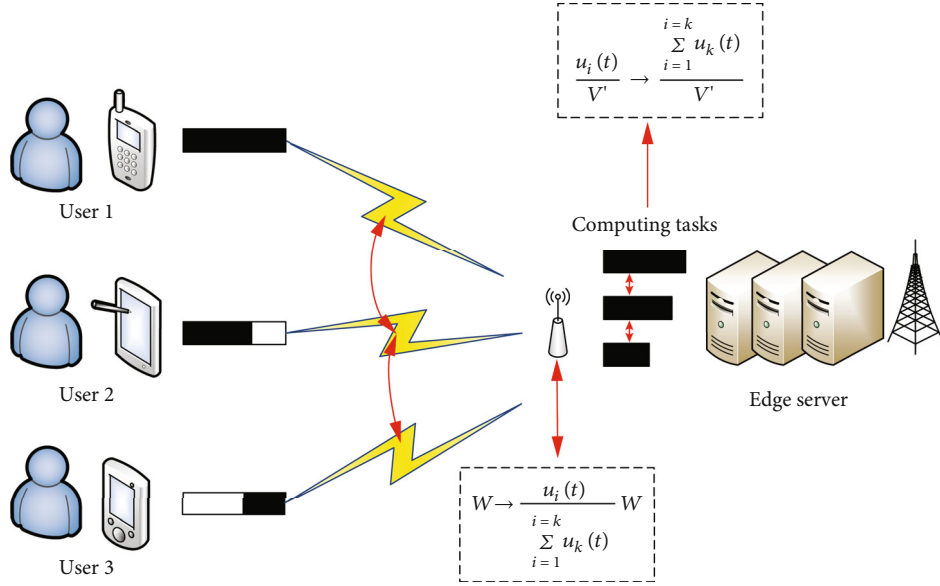


FIGURE 4: Competition in multiuser offloading.

common-state function $\Lambda(t)$ exists and satisfies the following relation [25]:

$$u_i^*(t) = \arg \min_{u_i(t)} H_i(t), \quad (10)$$

$$\dot{\Lambda}(t) = -\frac{\partial H_i(t)}{\partial x(t)}, \quad (11)$$

where $\Lambda(t)$ is a common-state function associated with QoE index trajectory $x(t)$. Using the common-state function $\Lambda(t)$, we can combine the benefits of edge users with the QoE level state to construct a Hamiltonian operator $H_i(t)$. Based on the principle of the Bellman dynamic programming, the constructed Hamiltonian operator can be described as follows [26, 27]:

$$H_i(t) = G_i(t) + \Lambda(t) \cdot \frac{dx(t)}{dt}. \quad (12)$$

According to the above, we need to find the optimal task quantity $u_i^*(t)$. And in order to get an open-loop Nash solution, we need to solve for the Hamiltonian given in (12). We will use the following lemma to solve it.

Lemma 2. *The optimal task quantity is*

$$u_i^*(t) = \frac{V' W \log_2(1 + (p/N)) [(R/y_i) - \chi x(t) + \varphi \Lambda(t)]}{r(t) [W \log_2(1 + (p/N)) + V'] (k+1)}. \quad (13)$$

Proof. The Hamiltonian system of edge devices can be expressed as

$$H_i(t) = G_i(t) + \Lambda(t) \cdot \frac{dx(t)}{dt}, \quad (14)$$

$$H_i(t) = \frac{R}{y_i} u_i(t) - \chi u_i(t) x(t) - r(t) u_i(t) \cdot \left[\frac{\sum_{k=1}^{k=i} u_k(t)}{V'} + \frac{u_i(t)}{(u_i(t) / \sum_{k=1}^{k=i} u_k(t)) W \times \log_2(1 + (p/N))} \right] + \Lambda(t) \left[\lambda x(t) + \varphi \sum_i u_i(t) \right]. \quad (15)$$

By calculating the partial derivative of the above equation with respect to $u_i(t)$, we can get

$$\frac{\partial H_i(t)}{\partial u_i(t)} = \frac{R}{y_i} - \chi x(t) - r(t) \left[u_i(t) + \sum_{i=1}^k u_i(t) \right] \cdot \left[\frac{1}{V'} + \frac{1}{W \log_2(1 + (p/N))} \right] + \varphi \Lambda(t). \quad (16)$$

Pay attention to $u_i(t) \sum_{k=1}^{k=i} u_k(i)$ in formula (15). Take the partial derivative of $u_i(t)$, and we can get

$$u_i(t) + \sum_{i=1}^k u_i(t). \quad (17)$$

For formula (16), let us make

$$C_1 = \frac{R}{y_i} - \chi x(t) + \varphi \Lambda(t), \quad (18)$$

$$C_2 = -r(t) \left[\frac{1}{V'} + \frac{1}{W \log_2(1 + (p/N))} \right]. \quad (19)$$

So we can get

$$\frac{\partial H_i(t)}{\partial u_i(t)} = C_2 \left[u_i(t) + \sum_{i=1}^k u_i(t) \right] + C_1. \quad (20)$$

Set formula (20) equal to 0 and perform the following transformation:

$$\frac{\partial H_i(t)}{\partial u_i(t)} = kC_2 \left[u_i(t) + \sum_{i=1}^k u_i(t) \right] + kC_1 = 0. \quad (21)$$

That is,

$$\frac{\partial H_i(t)}{\partial u_i(t)} = C_2 \left[\sum_{i=1}^k u_i(t) + k \sum_{i=1}^k u_i(t) \right] + kC_1 = 0. \quad (22)$$

Then, we can get

$$(k+1)C_2 \sum_{i=1}^k u_i(t) = -kC_1. \quad (23)$$

That is,

$$\sum_{i=1}^k u_i(t) = \frac{-kC_1}{C_2(k+1)}. \quad (24)$$

Formulas (20) and (24) can be obtained as follows:

$$u_i(t) = \frac{-C_1}{C_2(k+1)}. \quad (25)$$

Combining formulas (18), (19), and (25), we get the optimal task quantity as follows:

$$u_i^*(t) = \frac{V' W \log_2(1 + (p/N)) [(R/y_i) - \chi x(t) + \varphi \Lambda(t)]}{r(t) [W \log_2(1 + (p/N)) + V'] (k+1)}. \quad (26)$$

Lemma 2 is proven. \square

We find that the expression of the optimal task quantity contains the common-state function $\Lambda(t)$, so we need to find the relevant expression form of the common-state function $\Lambda(t)$.

According to (11), we have

$$\dot{\Lambda}(t) = -\frac{\partial H_i(t)}{\partial x(t)}. \quad (27)$$

The partial derivative of $H_i(t)$ of the constructed Hamiltonian system with respect to $x(t)$ is obtained as follows:

$$\frac{d\Lambda(t)}{dt} = -\frac{\partial H_i(t)}{\partial x(t)} = \chi u_i(t) + \lambda \Lambda(t). \quad (28)$$

The optimal task quantity $u_i^*(t)$ obtained previously is substituted into the above equation and is coconstructed with $dx(t)/dt$ to form the differential equations as follows:

$$\begin{cases} u_i^*(t) = \frac{V' W \log_2(1 + (p/N)) [(R/y_i) - \chi x(t) + \varphi \Lambda(t)]}{2r(t) [W \log_2(1 + (p/N)) + V'] (k+1)}, \\ \frac{d\Lambda(t)}{dt} = -\frac{\partial H_i(t)}{\partial x(t)} = \chi u_i^*(t) + \lambda \Lambda(t), \\ dx^*(t) = \left[\lambda x^*(t) + \varphi \sum_i u_i^*(t) \right] dt. \end{cases} \quad (29)$$

Solve the differential equations to solve the common mode function $\Lambda(t)$, $u_i^*(t)$, and $x^*(t)$ trajectory form.

5.2. Feedback Nash Solution of the Game. In the previous section, we obtain the Nash equilibrium solution of the open loop. The optimal strategy of each user is only dependent on the current time and start state of the game, which is enough for some analysis and application. But to avoid the time consistency problem, we will find the feedback Nash equilibrium solution in this section.

We presume that the strategy set $\{u_i^*(t) = u_i^*(\bullet), i \in N\}$ forms the Nash equilibrium solution of the feedback for equations (3) and (7) in the finite time domain if and only if there exists a differential equation that is continuously and differentiable $V_t^i(x, t): R^{\omega} \rightarrow R$, and the following Behrman equation satisfies [26]

$$\begin{aligned} -V_t^i(x, t) = \max_{u_i(t)} \left\{ e^{-\delta(t-t_0)} \left\{ R \frac{u_i(t)}{y_i} - \chi u_i(t) x(t) - r(t) u_i(t) \right. \right. \\ \left. \left. \cdot \left[\frac{\sum_{k=1}^{k=i} u_k(t)}{V'} + \frac{u_i(t)}{(u_i(t) / \sum_{k=1}^{k=i} u_k(t)) W \times \log_2(1 + (p/N))} \right] \right\} \right\} \\ + V_x^i(x, t) \left[\lambda x(t) + \varphi \times \sum_i u_i(t) \right]. \end{aligned} \quad (30)$$

And when $t = T$,

$$V^i(x, T) = e^{-\delta(T-t_0)} q x(T). \quad (31)$$

We use Lemma 3 to solve the Bellman equation (19) and obtain the optimal strategy $u_i^*(t)$.

Lemma 3. *The optimal task quantity is*

$$u_i^*(t) = \frac{e^{-\delta(t-t_0)} V' W \log_2(1 + (p/N)) [(R/y_i) - \chi x(t) + \varphi e^{-\delta(t-t_0)} \Lambda(t)]}{r(t) [W \log_2(1 + (p/N)) + V'] (k+1)}. \quad (32)$$

TABLE 1: Parameter setting in simulation and experiment for Figure 5.

Parameters	k	χ	R	V'	W	p	N	y_i	ϕ	λ
Values	1	0.05	100	50	5	0.05	50	0.5	0.75	0.8
	2	0.05	100	50	5	0.05	50	0.5	0.75	0.8
	3	0.05	100	50	5	0.05	50	0.5	0.75	0.8
	5	0.05	100	50	5	0.05	50	0.5	0.75	0.8
	7	0.05	100	50	5	0.05	50	0.5	0.75	0.8

Proof. If formula (19) derives $u_i(t)$ and makes it equal to 0, we can get

$$u_i^*(t) = \frac{e^{-\delta(t-t_0)} V' W \log_2(1 + (p/N)) [(R/y_i) - \chi x(t) + \varphi V_x^i(x, t)]}{r(t) [W \log_2(1 + (p/N)) + V'] (k+1)}. \quad (33)$$

According to formula (21), we need to find the specific expression of $V_x^i(x, t)$ or the relationship with $u_i^*(t)$.

Considering the characteristics of differential equation (19) and the form of solution, we use the form of $V^i(x, t) = Ax + B$ to solve, and that is

$$V^i(x, t) = e^{-\delta(t-t_0)} [A(t)x + B(t)], \quad (34)$$

where δ is the discount rate.

We can obtain the partial derivatives of t and x for formula (22):

$$\begin{aligned} V_t^i(x, t) &= e^{-\delta(t-t_0)} [-\delta A(t) + \dot{A}(t)] x + e^{-\delta(t-t_0)} [-\delta B(t) + \dot{B}(t)], \\ V_x^i(x, t) &= e^{-\delta(t-t_0)} A(t). \end{aligned} \quad (35)$$

Combining formulas (19), (23), and (24) and according to the equation on the left and right sides of the corresponding term coefficients which are equal, we can get

$$\begin{aligned} A'(t) &= (\lambda - \delta)A(t) + \chi u_i^*(t), \\ B'(t) &= \delta B(t) - R \frac{u_i^*(t)}{y_i} + r(t) u_i^*(t) \\ &\quad \cdot \left[\frac{\sum_{k=1}^{k=i} u_k(t)}{V'} + \frac{u_i^*(t)}{(u_i^*(t) / \sum_{k=1}^{k=i} u_k(t)) W \times \log_2(1 + (p/N))} \right] \\ &\quad - A(t) \varphi \sum_i u_i^*(t). \end{aligned} \quad (36)$$

According to formulas (22) and (25), we can get

$$u_i^*(t) = \frac{e^{-\delta(t-t_0)} V' W \log_2(1 + (p/N)) [(R/y_i) - \chi x(t) + \varphi e^{-\delta(t-t_0)} A(t)]}{r(t) [W \log_2(1 + (p/N)) + V'] (k+1)}. \quad (37)$$

Lemma 3 is proven. \square

TABLE 2: Parameter setting in simulation and experiment for Figure 6.

Parameters	k	χ	R	V'	W	p	N	y_i	ϕ	λ
Values	7	0.05	100	50	5	0.05	50	0.15	0.75	0.8
	7	0.05	100	50	5	0.05	50	0.25	0.75	0.8
	7	0.05	100	50	5	0.05	50	0.3	0.75	0.8
	7	0.05	100	50	5	0.05	50	0.5	0.75	0.8
	7	0.05	100	50	5	0.05	50	0.7	0.75	0.8

According to Lemma 3, we can get the following differential equations:

$$\begin{cases} u_i^*(t) = \frac{e^{-\delta(t-t_0)} V' W \log_2(1 + (p/N)) [(R/y_i) - \chi x(t) + \varphi e^{-\delta(t-t_0)} A(t)]}{r(t) [W \log_2(1 + (p/N)) + V'] (k+1)}, \\ A'(t) = (\lambda - \delta)A(t) + \chi u_i^*(t), \\ dx^*(t) = \left[\lambda x^*(t) + \varphi \sum_i u_i^*(t) \right] dt. \end{cases} \quad (38)$$

We will use MATLAB in the next section to numerically simulate and solve the above equations.

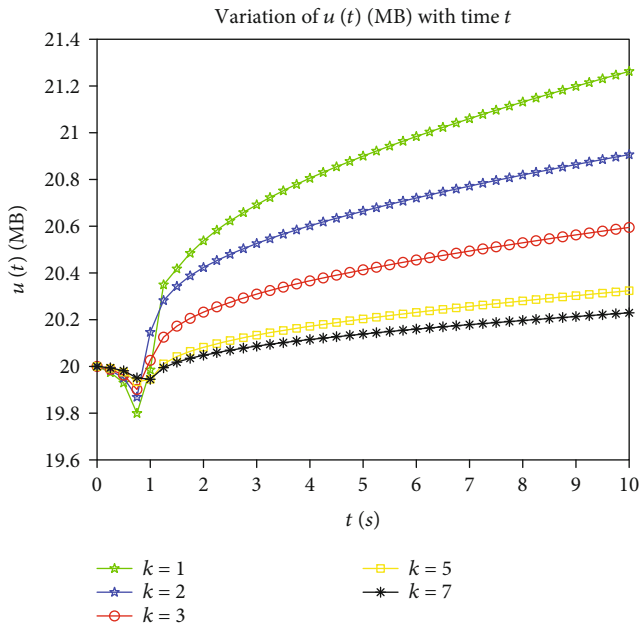
6. Results

We will numerically simulate the Nash equilibrium solution of the open loop and the Nash equilibrium solution of the feedback, obtained in the previous section under the optimal task offloading amount $u(t)$, QoE level $x(t)$, and user income $G_i(t)$ in this section. Through the comparison of the changes of the two parameters of the number of users k and the uninstall percentage y_i of the total task volume, the analysis and discussion are carried out with the above quantities. According to [28], the simulation setting of this paper is MATLAB 2018b in Windows 10, and the simulation settings are shown in Tables 1 and 2.

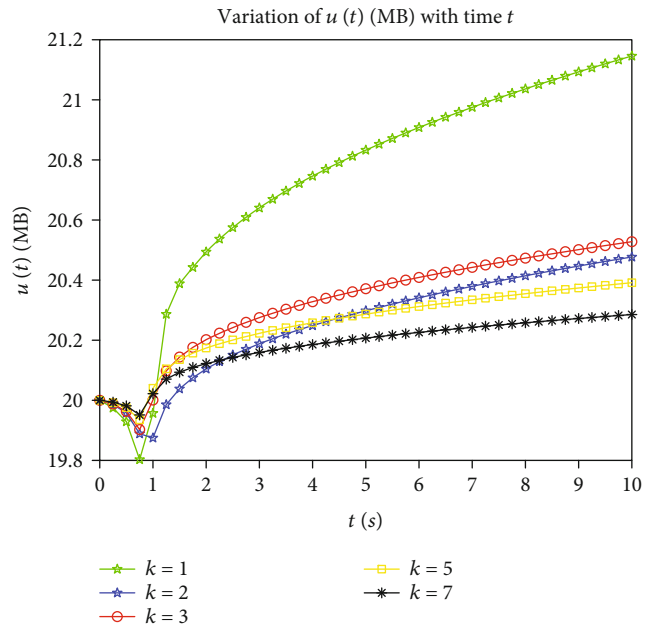
As shown in the table, the value of V' refers to [29]. According to the peak value and base value given by [29], calculate the mean value.

As shown in Figure 5, it shows the changing trend of $u(t)$, $x(t)$, and $G(t)$ during the game time; at the same time, we use different numbers of users k to compare and analyze the impact of offloading when multiple users are offloaded.

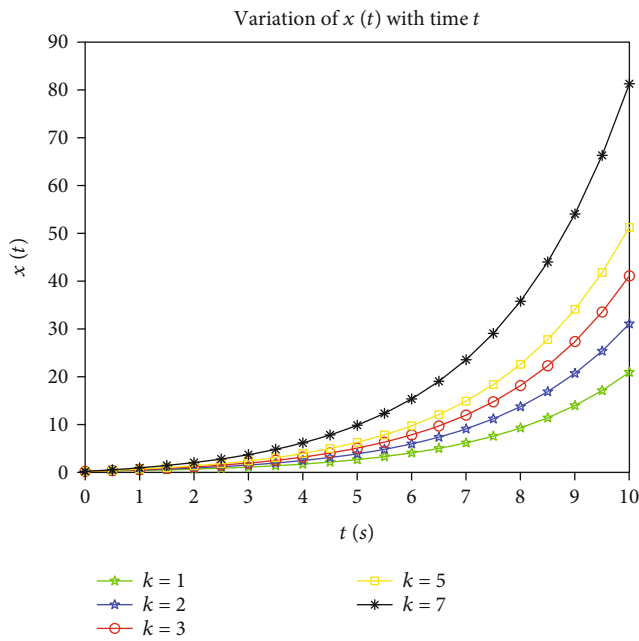
In Figures 5(a) and 5(b), we can find that with the growth of the number of edge users k , the optimal amount



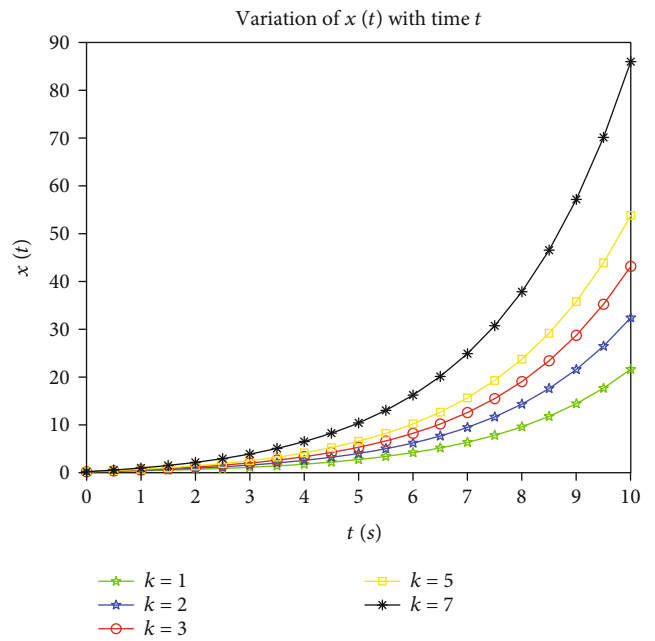
(a) Open loop



(b) Feedback



(c) Open loop



(d) Feedback

FIGURE 5: Continued.

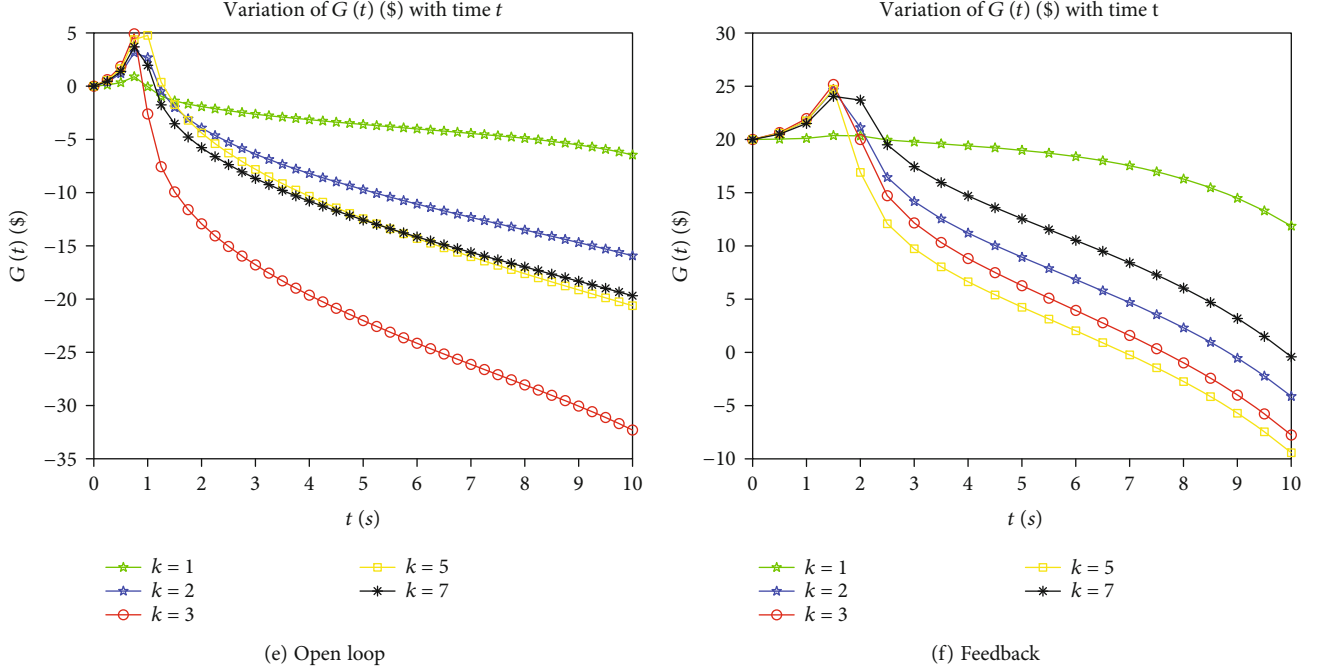


FIGURE 5: Variation of $u(t)$, $x(t)$, and $G(t)$ with time t when $y_i = 0.5$.

of offloading tasks for edge users is decreasing, because in the many-to-one environment, the processing capacity of the edge server has an upper limit. In order to ensure better service performance, edge servers usually reserve a part of the CPU cache to ensure uninterrupted task supply when they perform offloading tasks.

In Figures 5(c) and 5(d), we can find that as k increases, the QoE level $x(t)$ of the entire edge system increases. This is because although the increase in the number of users will intensify the competition among users and lead to a decrease in task offloading, the QoE level of the system as a whole will increase. It is worth noting that the value of the QoE index φ here is greater than 0, indicating that the increase in the number of users has not reached or exceeded the processing upper limit of the edge server. If the number of users exceeds a certain threshold (e.g., a DDOS attack occurs), the task requests of edge users will become invalid, which will lead to a sharp drop in the QoE level.

In Figures 5(e) and 5(f), we can find that as k increases, the profit $G(t)$ of edge users decreases. This is due to the growth of the number of users caused the decrease in the amount of offloading tasks, and the decrease in the amount of offloading tasks leads to the increase in tasks in local processing. Local processing will increase the loss of the equipment itself, so the income will be reduced.

In Figure 5, we find that the optimal decision quantity of the two solutions may be slightly different, but the trajectories of the two groups of results are roughly the same. This is because in the open-loop solution, the participants only care about the current moment and the future moment of income, while in the feedback solution, the participants also care about the overall income at the end of the game. This

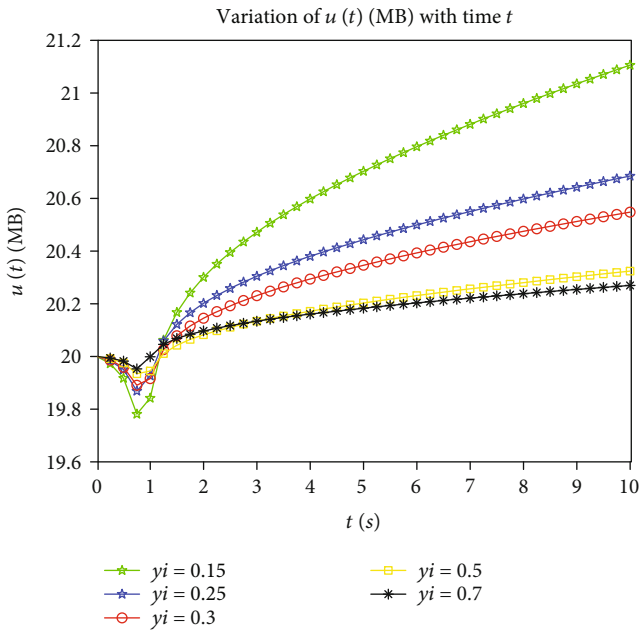
also leads to the difference in the final benefits between the two solutions.

In addition, we also find that when the user income is high, the QoE level is low, so in practical application, it should be balanced dynamically to carry out.

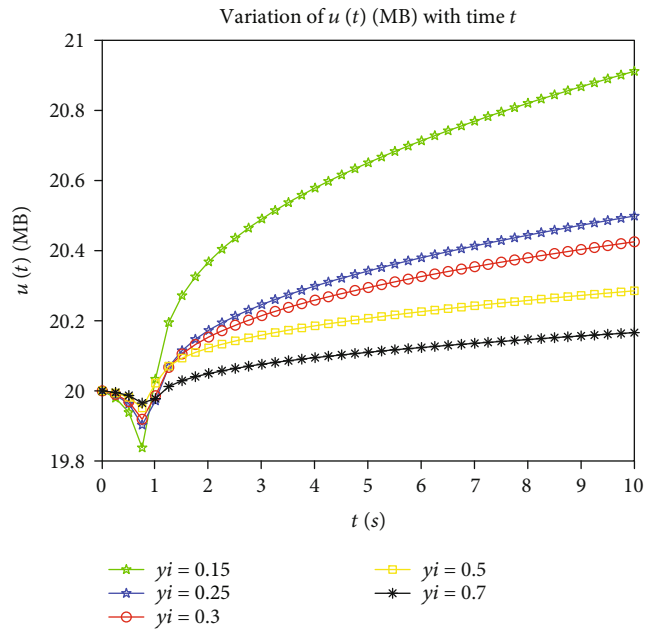
As shown in Figure 6, it shows the changing trend of $u(t)$, $x(t)$, and $G(t)$ during the game time; at the same time, we use different offloading proportions y_i for comparison to analyze the relationship between the optimal task offloading decision volume and the revenue of different edge users according to the offloading proportions.

We can find from Figures 6(a) and 6(b) that the optimal amount of offloaded tasks decreases with the increase in y_i . This is because when the total amount of tasks $U_i(t)$ is determined, the amount of offloaded tasks is determined according to the size of y_i . However, in order to maintain its own revenue, the smaller the y_i is, the greater the amount of task offloading will be required and vice versa. According to Figures 6(c) and 6(d), we can find that the QoE level $x(t)$ decreases with the increase in offloading component y_i , which is due to a large amount of offloading and the large time delay. Therefore, the QoE obtained by most tasks completed by offloading is smaller than that obtained by local completion. However, according to Figures 6(e) and 6(f), although the QoE obtained when the task is completed locally is higher, the benefit $G(t)$ is smaller.

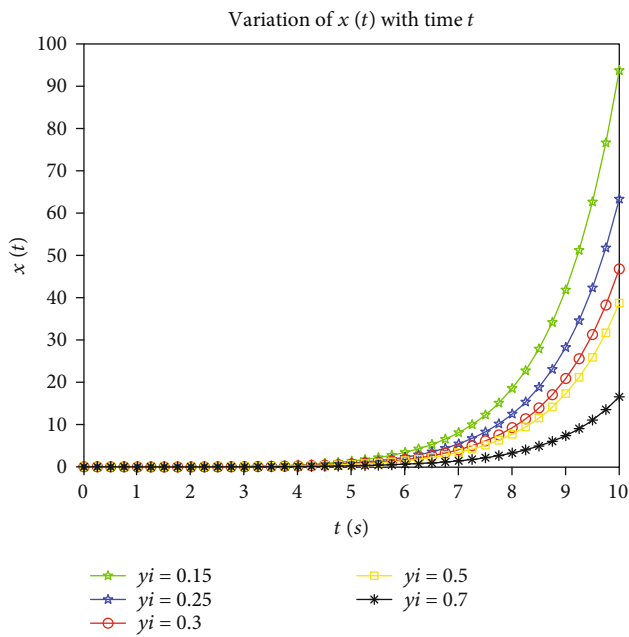
This is not difficult to explain because the delay of processing tasks locally is much less than that of offloading tasks to edge servers (especially those deployed on LEO satellites), so user satisfaction decreases as the offloading ratio y_i increases; at the same time, there is a decrease in satisfaction but an increase in revenue, because too much local processing leads to an increase in self-consumption and thus a



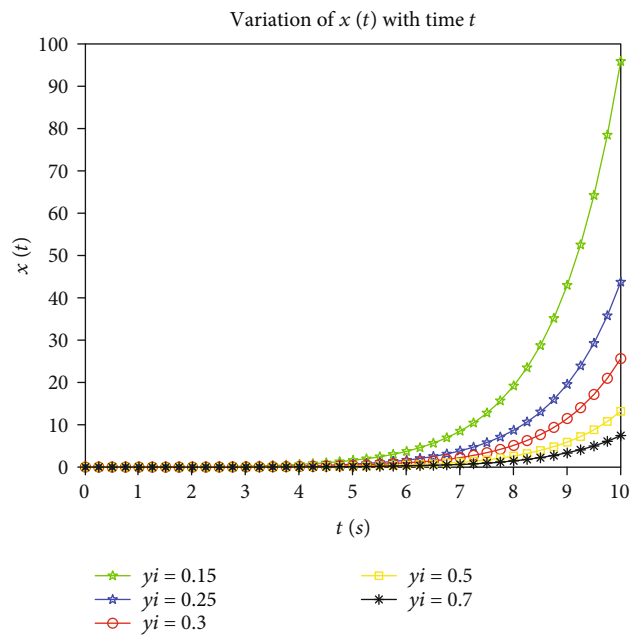
(a) Open loop



(b) Feedback



(c) Open loop



(d) Feedback

FIGURE 6: Continued.

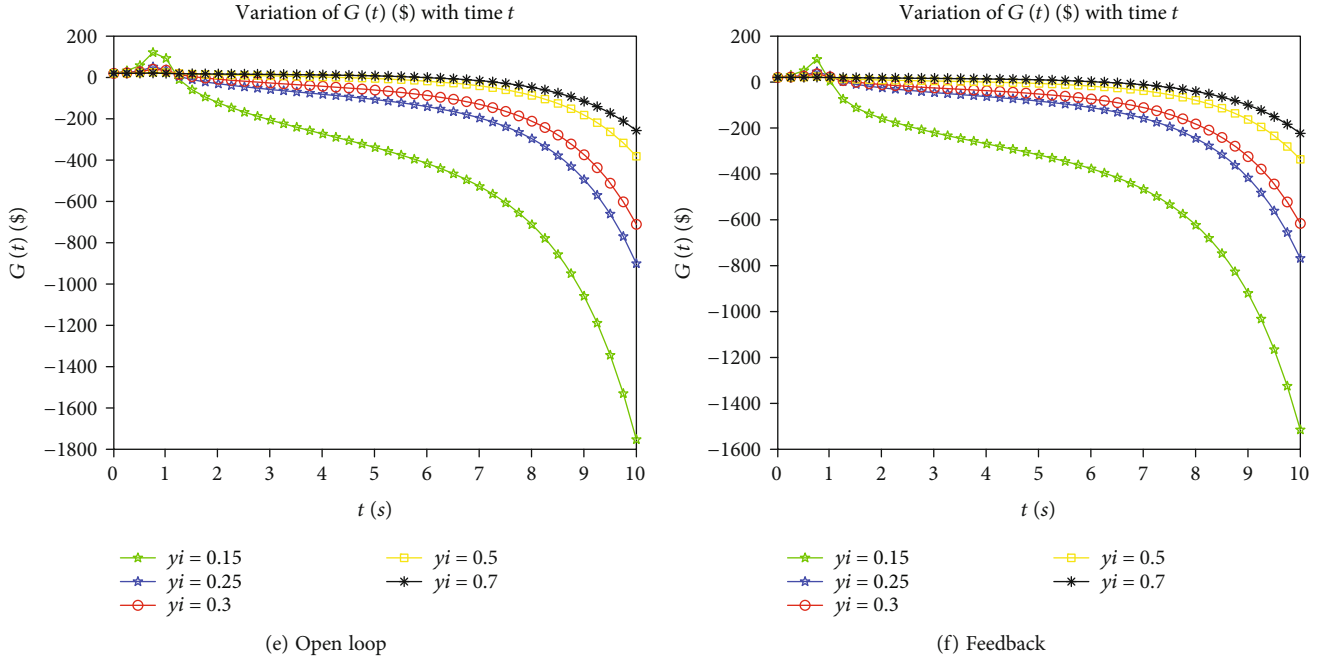


FIGURE 6: Variation of $u(t)$, $x(t)$, and $G(t)$ with time t when $k = 7$.

decrease in revenue. However, the decrease in satisfaction indirectly affects the revenue of the edge server. Therefore, how to dynamically balance the relationship between the two needs to start from the specific scene requirements to achieve the optimal system state of each other.

7. Conclusion

In this paper, LEO-MEC multiaccess converged network architecture is established to solve the problem of task offloading for edge users with a limited geographical environment, and a differential game model is established for the decision of task offloading in the architecture. The problem of mutual competition when multiple users offload tasks is discussed. By solving the Nash equilibrium solution of the open loop and the Nash equilibrium solution of the feedback, we get the optimal amount of task offloading $u_i^*(t)$. Finally, the relationship between $u(t)x(t)G(t)$ and the number of edge users k and the offloading component y_i is discussed by numerical simulation, which plays a certain role in balancing the user income and QoE level in practical work.

In the future work, we will continue to explore the space network architecture of LEO satellite and edge computing fusion. Due to the low orbit altitude of LEO satellites, the satellites move fast, which brings a series of challenges, such as Doppler frequency shift and short radiation time of satellites. We will also discuss the allocation of computational offloading resources in the “bent-tube” mode scenario where the LEO satellite is transmitted as a relay node.

Data Availability

The data are available at <http://www.spec.org/cpu2006/results/cpu2006.html>.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] E. Ahmed, A. Gani, M. Khurram Khan, R. Buyya, and S. U. Khan, “Seamless application execution in mobile cloud computing: motivation, taxonomy, and open challenges,” *Journal of Network and Computer Applications*, vol. 52, pp. 154–172, 2015.
- [2] Q. Tang and B. Li, “Overview of mobile edge computing in space-air-ground intergrated network,” *Radio.Communications Technology*, vol. 47, no. 1, pp. 27–35, 2021.
- [3] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, “The case for vm-based cloudlets in mobile computing,” *IEEE pervasive Computing*, vol. 8, no. 4, pp. 14–23, 2009.
- [4] W. L. Zhang, B. Guo, Y. Shen, Y. Wang, W. Xiong, and L. Duan, “Computation offloading on intelligent mobile terminal,” *Chinese Journal of Computers*, vol. 39, no. 5, pp. 1021–1038, 2016.
- [5] J. Zhang, W. Xia, Z. Cheng et al., “An evolutionary game for joint wireless and cloud resource allocation in mobile edge computing,” in *2017 9th International Conference on Wireless Communications and Signal Processing*, pp. 1–6, Nanjing, China, Oct 11-13, 2017.
- [6] X. Chen, L. Jiao, W. Li, and X. Fu, “Efficient multi-user computation offloading for mobile-edge cloud computing,” *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, 2016.
- [7] M. H. Chen, B. Liang, and M. Dong, “Joint offloading decision and resource allocation for multi-user multi-task mobile cloud,” *2016 IEEE international conference on communications*, 2016, pp. 1–6, Kuala Lumpur, Malaysia, may 22-27, 2016.

- [8] Q. V. Pham, T. Le Anh, N. H. Tran, and C. S. Hong, "Decentralized computation offloading and resource allocation in heterogeneous networks with mobile edge computing," 2018, arXiv preprint arXiv:1803.00683.
- [9] K. Zhang, Y. Mao, S. Leng et al., "Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks," *IEEE Access*, vol. 4, pp. 5896–5907, 2016.
- [10] P. Zhao, H. Tian, C. Qin, and G. Nie, "Energy-saving offloading by jointly allocating radio and computational resources for mobile edge computing," *IEEE Access*, vol. 5, pp. 11255–11268, 2017.
- [11] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3590–3605, 2016.
- [12] S. Ahn, J. Lee, S. Park, S. H. S. Newaz, and J. K. Choi, "Competitive partial computation offloading for maximizing energy efficiency in mobile cloud computing," *IEEE Access*, vol. 6, pp. 899–912, 2018.
- [13] X. Tao, K. Ota, M. Dong, H. Qi, and K. Li, "Performance guaranteed computation offloading for mobile-edge cloud computing," *IEEE Wireless Communications Letters*, vol. 6, no. 6, pp. 774–777, 2017.
- [14] Y. Dai, D. Xu, S. Maharjan, and Y. Zhang, "Joint computation offloading and user association in multi-task mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 12, pp. 12313–12325, 2018.
- [15] M. Deng, H. Tian, and B. Fan, "Fine-granularity based application offloading policy in cloud-enhanced small cell networks," in *2016 IEEE international conference on communications workshops (ICC)*, pp. 638–643, Kuala Lumpur, Malaysia, May 2016.
- [16] Y. Wang, J. Zhang, X. Zhang, P. Wang, and L. Liu, "A computation offloading strategy in satellite terrestrial networks with double edge computing," in *2018 IEEE international conference on communication systems (ICCS)*, pp. 450–455, Chengdu, China, December 2018.
- [17] Z. Zhang, W. Zhang, and F. H. Tseng, "Satellite mobile edge computing: improving QoS of high-speed satellite-terrestrial networks using edge computing techniques," *IEEE Network*, vol. 33, no. 1, pp. 70–76, 2019.
- [18] C. Qiu, H. Yao, F. R. Yu, F. Xu, and C. Zhao, "Deep q-learning aided networking, caching, and computing resources allocation in software-defined satellite-terrestrial networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 6, pp. 5871–5883, 2019.
- [19] A. Alsharoa and M. S. Alouini, "Improvement of the global connectivity using integrated satellite-airborne-terrestrial networks with resource optimization," *IEEE Transactions on Wireless Communications*, vol. 19, no. 8, pp. 5088–5100, 2020.
- [20] F. Xu, F. Yang, C. Zhao, and S. Wu, "Deep reinforcement learning based joint edge resource management in maritime network," *China Communications*, vol. 17, no. 5, pp. 211–222, 2020.
- [21] K. Y. Zhang, X. L. Gui, D. W. Ren, J. Li, J. Wu, and D. S. Ren, "Review of computing migration and content caching in mobile edge networks," *Journal of Software*, vol. 8, pp. 2491–2516, 2019.
- [22] H. Li, H. Xu, C. Zhou, X. Lu, and Z. Han, "Joint optimization strategy of computation offloading and resource allocation in multi-access edge computing environment," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 9, pp. 10214–10226, 2020.
- [23] I. Rufus, *Differential Games III: The Basic Principles of the Solution Process*, RAND CORP SANTA MONICA CA, Mineola, NY, USA, 1954.
- [24] Y. Luo, *Research on QoE of Mobile Internet Service*, Beijing University of Posts and telecommunications, 2013.
- [25] F. Peng, *Research on Edge Computing Network and QoE Aware Resource Scheduling*, Beijing University of Posts and telecommunications, 2019.
- [26] H. Xu, W. Huang, Y. Zhou, D. Yang, M. Li, and Z. Han, "Edge computing resource allocation for unmanned aerial vehicle assisted mobile network with blockchain applications," *IEEE Transactions on Wireless Communications*, vol. 20, no. 5, pp. 3107–3121, 2021.
- [27] H. Wang, L. Wang, Z. Zhou, X. Tao, G. Pau, and F. Arena, "Blockchain-based resource allocation model in fog computing," *Applied Sciences*, vol. 9, no. 24, p. 5538, 2019.
- [28] H. Xu, C. Guo, and L. Zhang, "Optimal power control in wireless powered sensor networks: a dynamic game-based approach," *Sensors*, vol. 17, no. 3, p. 547, 2017.
- [29] "ALL SPEC CPU2006 results6–142014, <http://www.spec.org/cpu2006/results/cpu2006.html>.