IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received December 5, 2019, accepted December 26, 2019, date of publication December 30, 2019,
date of current version January 21, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2963068

A Game-Theoretic Approach to Computation Offloading in Satellite Edge Computing

YUXUAN WANG, JUN YANG, XIYE GUO, AND ZHI QU
National University of Defense Technology, Changsha 410073, China

Corresponding author: Jun Yang (john323@163.com)

ABSTRACT Mobile edge computing (MEC) is proposed as a new paradigm to meet the ever-increasing computation requirements, which is caused by the rapid growth of the Internet of Things (IoT) devices. As a supplement to the terrestrial network, satellites can provide communication to terrestrial devices in some harsh environments and natural disasters. Satellite edge computing is becoming an emerging topic and technology. In this paper, a game-theoretic approach to the optimization of computation offloading strategy in satellite edge computing is proposed. The system model for computation offloading in satellite edge computing is established, considering the intermittent terrestrial-satellite communication caused by satellites orbiting. We conduct a computation offloading game framework and compute the response time and energy consumption of a task based on the queuing theory as metrics of optimizing performance. The existence and uniqueness of the Nash equilibrium is theoretically proved, and an iterative algorithm is proposed to find the Nash equilibrium. Simulation results validate the proposed algorithm and show that the game-based offloading strategy can greatly reduce the average cost of a device.

INDEX TERMS Edge computing, game theory, Nash equilibrium, offloading strategy optimization, queuing system.

I. INTRODUCTION

In recent years, relying on the construction of infrastructure (such as the terrestrial Internet and mobile network) and the popularization of smart devices, IoT technology has developed rapidly. According to the survey of Gartner Inc., it is estimated that the total number of networked or connected IoT devices in 2020 will be 20.8 billion [1]. Limited by cost and technology, the terrestrial network covers only about 20% of the total land area and is mainly concentrated in urban areas. For some harsh environments such as deserts, forests, mountains, and oceans, the terrestrial network cannot cover entirely. Besides, in case of natural disasters, such as floods, earthquakes, tsunamis, etc., the terrestrial network is vulnerable. With its extensive coverage and system robustness, satellite communication systems can provide access services for IoT devices in remote areas, realizing the ''Internet of Everything'' in the real sense of the world. Satellites have not only become an important part of the Internet of Things, but also a powerful complement to future 5G/6G communication [2]–[4].

The associate editor coordinating the review of this manuscript and approving it for publication was Charith Perara.

The computing and energy resources of devices are usually limited. Thus, IoT devices need to rely on the cloud to store and process data. However, cloud computing platforms are often physically and logically distant from the terminal. The proliferation of devices and associated data streams has put significant pressure on the network. It becomes a bottleneck of providing satisfactory quality of service (QoS). MEC offers a new paradigm for a myriad of mission-critical applications [5]. The core idea is to extend the capabilities of the cloud to the network edge, closer to IoT devices, to reduce data traffic and response latency. Comparing to cloud computing, MEC has the advantage of significantly reducing latency, avoiding congestion, and prolonging the battery life of devices [6], [7]. Therefore, it has recently been widely used in both industry and academia [8]–[10]. Recently, some studies have combined satellite with edge computing to deploy MEC servers on satellites for lower latency and more general-purpose applications [11]–[15]. Satellite edge computing is becoming an emerging topic and technology.

As one of the principal challenges of MEC, the computation offloading, i.e., the problem of transmitting computation tasks from mobiles to MEC servers has been studied in various application scenarios. When mobile devices aggressively and unrestrictedly offload their computation tasks to the MEC server, it usually causes an enormous communication burden and severe signal crosstalk to the network. Meanwhile, it will overload the MEC server, and significantly increase the overall task processing time. It will compromise the benefits of computation offloading. Therefore, the strategy of computation offloading needs to be optimized to obtain better QoS. However, to the best of the author's knowledge, there has been no research on computation offloading optimization in satellite edge computing. Different from the fixation of base stations in MEC, satellites orbit and have high dynamics, which leads to an intermittent terrestrial-satellite communication.

In this work, we propose a game-theoretic approach to the optimization of computation offloading strategy for terrestrial devices in satellite edge computing scenarios. Our main contributions are summarized as follows.

1) We establish the system model of computation offloading in satellite edge computing. There are multiple terrestrial devices located in the same area, where computation tasks can be executed locally on the device or be offloaded to satellites for execution. Unlike other MEC scenarios, the intermittent terrestrial-satellite communication caused by satellite orbiting is considered in the system model.
2) We formulate an offloading game model, in which each device will selfishly choose the strategy that will minimize its cost. The response time and energy consumption of a task are computed based on the queuing theory. They are the metrics of optimizing performance. The existence and uniqueness of the Nash equilibrium is theoretically proved.
3) We propose an iterative algorithm to search the Nash equilibrium of the game. Simulation results validate the theoretical analysis and the convergence of the algorithm. The game-based offloading strategy can greatly reduce the average cost of a device.

The remainder of this paper is organized as follows. In Section II, a rapid overview of related work is illustrated. Therefore, the system model is established in Section III. The computation offloading game formulation and the proof of the existence and uniqueness of the Nash equilibrium are presented in Section IV. An iterative algorithm to find the Nash equilibrium is proposed in Section V. The simulation results are presented and discussed in Section VI. Finally, we conclude this work in Section VII.

II. RELATED WORKS

For all we know, there have been no researches on computation offloading in satellite edge computing. Therefore, we present some literature on optimizing the offloading strategy in MEC. The relevant technologies described are helpful for our research. For the most recent comprehensive survey, readers can refer to [16]–[18].

According to the number of users, computation offloading can be divided into two categories: single-user scenario and multi-user scenario. In a single-user scenario, the offloading strategy optimization problem is often converted into an optimal programming problem. Wang *et al.* developed a low-complexity adaptive offloading decision-transmission scheduling scheme based on the Lyapunov optimization theory for mobile devices, optimizing the average execution time and average energy consumption of tasks [19]. Liu *et al.* used the Markov decision process to develop a task offloading strategy with minimum delay under power constraints and proposed an effective one-dimensional search algorithm to find the optimal task scheduling strategy. This strategy has a shorter average execution delay than the baseline strategy [20]. Mao *et al.* proposed a Lyapunov optimization-based dynamic calculation offloading (LODCO) algorithm, which can jointly determine the task offloading strategy, CPU calculation frequency, and mobile device transmit power. Through the instantaneous auxiliary information, the optimal solution is obtained through the bisection search, which reduces the energy consumption of the device [21]. Zhang *et al.* proposed an Energy-Efficient Computation Offloading (EECO) scheme, which optimizes the offloading strategy and radio resource allocation, and achieves minimum energy consumption under delay constraints [22]. In general, for a single-user scenario, researchers have focused on optimizing task offloading strategies and radio resource allocation (such as channel, spectrum, transmit power, etc.). The optimization metrics are task latency and/or energy consumption.

In a multi-user scenario, computation offloading optimization is often modeled as a game problem. Li established the M/G/1 queuing model and the non-cooperative game framework for the multi-user, non-cooperative computation offloading scenario. Through theoretical calculations, the existence of the Nash equilibrium of the game is proved, and a distributed algorithm is designed to find the equilibrium [23]. Cardellini *et al.* describe the optimal computation offloading problem for non-cooperative users as a generalized Nash equilibrium problem (GNEP) for the device-edge-cloud three-tier architecture. The existence of the Nash equilibrium is theoretically proved, and the characteristics of equilibrium are illustrated by numerical examples [24]. Cao *et al.* demonstrated that the multi-user computing offload problem is a potential game, and there is at least one pure strategy Nash equilibrium. They proposed a fully distributed computation offloading (FDCO) algorithm based on machine learning technology, which can converge to purely strategic Nash equilibrium without any information exchange [25]. Zheng *et al.* studied the multi-user computation offloading problem in a dynamic environment and expressed the user's decision process as a stochastic game. They prove that the stochastic game is equivalent to a weighted potential game with at least one Nash equilibrium, and propose a multi-agent stochastic learning algorithm to search the Nash equilibrium with a guaranteed convergence rate [26]. In general, the game problem in multi-user scenarios focuses on the establishment of the game model, the selection of optimization targets, and the algorithm for finding Nash equilibrium. The optimization metric is similar to the single-user scenario, which is a combination of task response time and energy consumption.

III. SYSTEM MODEL

The satellite edge computing scenario is shown in Figure 1. A set of remote terrestrial devices is located in a small fixed area, and a set of satellites is in orbit. A task can be executed locally on a device or offloaded to an edge computing server deployed on a satellite via satellite-terrestrial communication. Due to the intermittent terrestrial-satellite communication, a device cannot offload tasks to a satellite at any time, i.e., tasks will be offloaded only if the satellite is flying over.

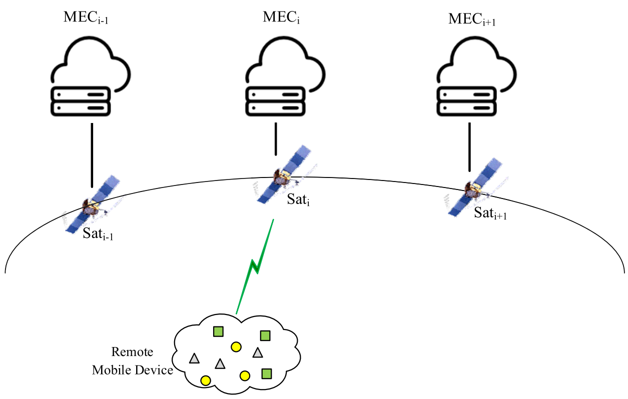


FIGURE 1. The satellite edge computing scenario.

We abstract the above satellite edge computing scenario to the system model shown in Figure 2. There are N mobile devices, denoted as $\mathbf{N} = \{1, 2, \ldots, N\}$, and M satellites covering the area, denoted as $\mathbf{M} = \{1, 2, \ldots, M\}$. Due to the limited onboard resources, we assume that only one edge computing server is deployed on satellite. The task execution on mobile devices and satellites is both characterized as a queuing system. The queuing theory has been widely used to analyze task processing and resource allocation in edge computing [27], [28]. Obviously, if all the tasks are executed locally on a device or offloaded to a satellite, the waiting time for execution and power consumption in the queue will increase dramatically. Therefore, the computation offloading strategy for each device should be optimized to improve the performance.

The system model consists of three parts, the orbit model of the satellite, the communication model of task offloading, and the computation model of task execution. For a better reading, the notations mainly used in this paper are summarized in Table 1.

A. ORBIT MODEL

Different from the stable communication in MEC, the satellite cannot always communicate with terrestrial devices. The terrestrial-satellite communication link can be established, only if the satellite orbits satisfy specific geometric constraints. The space geometry of the communication link between a satellite and a fixed location on the ground is shown in Figure 3.

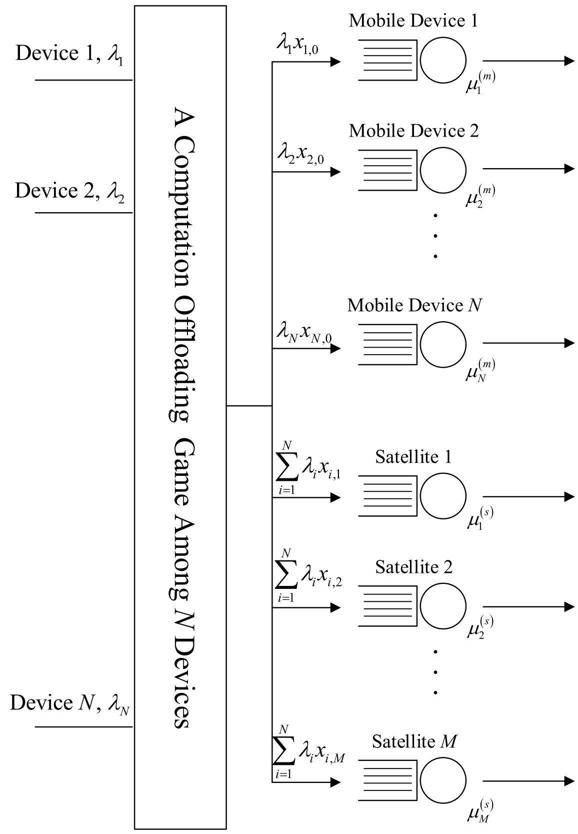


FIGURE 2. The system model of satellite edge computing.

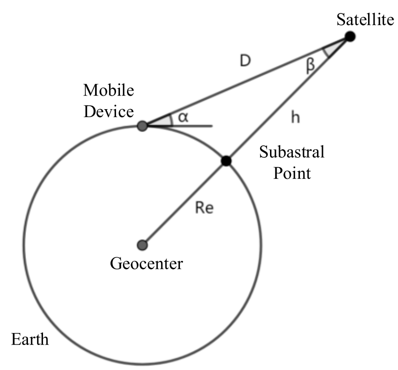


FIGURE 3. The space geometry of the link between a satellite and a mobile device.

Here, α is the elevation angle of the mobile device, β is the half-angle of view of the satellite, R_e is the radius of the earth, and h is the altitude of the satellite. Regardless of the influence of other factors, data transmission is only available when $\alpha > 0$. According to the geometric relationship, the expression of α can be obtained as follows [29]:

$$\alpha = \arctan \frac{\cos \Delta\phi \cos \varphi_t \cos \varphi_s + \sin \varphi_t \sin \varphi_s - \frac{R_e}{R_e + h}}{\sqrt{1 - (\sin \varphi_t \sin \varphi_s + \cos \Delta\phi \cos \varphi_t \cos \varphi_s)^2}} \quad (1)$$

Here, $\Delta\phi = \phi_t - \phi_s$, ϕ_t and φ_t are longitude and latitude of the mobile device, respectively. ϕ_s, and φ_s are longitude and latitude of the satellite, respectively.

TABLE 1. The notations are used in this paper.

Notation	Definition
N	the number of mobile devices
M	the number of satellites
α_j	elevation angle from the mobile device to satellite j
θ_j	the communication state with satellite j
T_j	the orbit period of satellite j
$R_{i,j}$	the uplink data rate for device i to satellite j
p_i	the transmit power of device i
$g_{i,j}$	the channel gain between device i and satellite j
σ_0	the background noise power
c_i	the number of computing resources required to execute a task generated by device i
d_i	the size of the computation input file of a task generated by device i
$C_i^{(m)}$	the computation power of device i
$C_j^{(s)}$	the computation power of satellite j
λ_i	the task generation rate of device i
$x_{i,0}$	the percentage of device i's tasks that are executed locally
$x_{i,j}$	the percentage of device i's tasks that are offloaded to satellite j
$\mathbf{x}_i$	the offloading strategy of device i
$\mathbf{x}_{-i}$	the strategies of all devices except i
$\mathbf{x}$	the offloading decision strategy for all devices
λ_j	$=\sum_{i=1}^{N} x_{i,j}\lambda_i$, the task arrival rate of satellite j
ρ_j	$=\lambda_j T_{i,j}^{sat}$, the utilization of the queue in satellite j.
$T_{i,j}$	the average response time of the device i's task offloaded on satellite j
$E_{i,j}$	average energy consumption of the device i's task offloaded on satellite j
$T_{i,0}^{loc}$	the average processing time on device i
$T_{i,0}^{wait}$	the average waiting time of the tasks on device i
$E_{i,0}$	the local average energy consumption of the task generated by device i
T_i	the average response time of all the device i's tasks
E_i	the average energy consumption of the task generated by device i
$P_i(\mathbf{x}_i, \mathbf{x}_{-i})$	the cost function of player i
G	$=\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N; P_1, P_2, \ldots, P_N\}$, the offloading game with N players
X_i	the set of strategies of device i
X	$=X_1\times X_2\times\ldots\times X_N$ the set of combination of all device's strategies
$\mathbf{x}^*$	$=\{\mathbf{x}_1^*, \mathbf{x}_2^*, \ldots, \mathbf{x}_N^*\}$, a pure strategy Nash equilibrium
ε	the accuracy requirement

If the distance between mobile devices is very far, the devices will communicate with different satellites. There is no competition in resources. Therefore, we assume that the terrestrial device is located in a small fixed area. The geometric relationship between these devices and a satellite is the same.

According to the positive or negative of the elevation angle α, we define the percentage of communication time in a cycle as $\boldsymbol{\theta} = \{\theta_1, \theta_2, \ldots, \theta_M\}$, $0 \le \theta_j \le 1$, where θ_j represents the percentage of time that device can communicate with satellite j in an orbit period.

B. COMMUNICATION MODEL

Suppose that mobile device i offloads its computation tasks to satellite j via a terrestrial-satellite network. We only consider that devices offload tasks to satellites. The results transmitted from the satellite to the mobile device is neglected in this work because the size of computation outcome data is much smaller than that of the computation input data [30]. Considering the mutual interference between the mobile device and background noise, the uplink data rate for mobile device i to satellite j can be calculated by [31]

$$R_{i,j} = B\log_2\left(1 + \frac{p_i g_{i,j}}{\sigma_0 + \sum_{s\in\mathbf{N}, s\neq i} p_s g_{s,j}}\right) \tag{2}$$

where B denotes the channel bandwidth, p_i denotes transmitting power of device i, $g_{i,j}$ denotes the channel gain between the device i and the satellite j, and σ_0 denotes the background noise power. According to (2), the data rate of task offloading is positively correlated with the transmission power of the device itself. However, the excessively high transmitting power leads to excessive energy consumption, which reduces the advantage of computation offloading. Additionally, due to the interference, if too many devices offload tasks to satellites, the data rate will decrease, leading to a long offloading time.

C. COMPUTATION MODEL

We assume that each mobile device could generate a series of homogeneous tasks. The tasks generated by device i can be represented by the resources required and the size of data, i.e., $\text{Task}_i = \{c_i, d_i\}$. Where c_i represents the number of computing resources required to execute a task; for example, c_i can be quantified by the number of CPU cycles. d_i denotes the size of the computation input file describing some information of a task, such as the program codes or the corresponding data. Both c_i and d_i are random variables, the means are $\bar{c}_i$ and $\bar{d}_i$, respectively, and second moments are $\overline{c_i^2}$ and $\overline{d_i^2}$, respectively. We assume that computation power (e.g. the CPU cycle/sec) of mobile device and satellite is $C_i^{(m)}$ and $C_j^{(s)}$ respectively, for all $1 \le i \le N$, $1 \le j \le M$.

Both terrestrial devices and satellites are characterized as an M/G/1 queuing system. In this way, the time interval for task generation follows the exponential distribution, and the task execution time follows an arbitrary probability distribution. We denote that device i generates task at rate λ_i, $1 \le i \le N$, i.e., the time interval for task generation is an independent and identically distributed (i.i.d.) random variable subject to the exponential distribution with a mean of $1/\lambda_i$. The percentage of device i's tasks that are executed locally and offloaded to a satellite is defined as the computation offloading strategy of device i, denoted by $\mathbf{x}_i = \{x_{i,0}, x_{i,1}, \ldots, x_{i,M}\}$. Here, $x_{i,0}$ represents the percentage of the tasks that are executed locally, and $x_{i,j}$ represents the percentage of the tasks that are offloaded to satellite j. Obviously, the offloading strategy for any device meets the constraints below.

$$\begin{aligned} x_{i,j} &\ge 0, \quad \forall i \in \mathbf{N},\ \forall j \in \mathbf{M} \\ \sum_{j=0}^{M} x_{i,j} &= 1, \quad \forall i \in \mathbf{N} \end{aligned} \tag{3}$$

IV. PROBLEM FORMULATION AND ANALYSIS

In satellite edge computing, mobile devices compete for computing and communication resources. The mobile devices are selfish and competitive to choose the offloading strategy, which is most beneficial to them. Therefore, we formulate the computation offloading problem as a game problem. Each player (i.e., mobile device) expects to get a better QoS by deciding whether and where to offload a task.

A. COMPUTATION OFFLOADING GAME

The strategy of player i is the percentage of tasks that are executed locally on a device or offloaded to satellite, denoted by $\mathbf{x}_i = \{x_{i,0}, x_{i,1}, \ldots, x_{i,M}\} \in X_i$, as described in Section III. Here, X_i, which is the set of strategies of device i, is closed and convex (because of $X_i \subseteq \mathbb{R}^M$ and $x_{i,0} + x_{i,1} + \ldots + x_{i,M} = 1$). Let $X = X_1 \times X_2 \times \ldots \times X_N$ be the set of combinations of all devices' strategies. Denoted by $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\} \in X$ the overall vector of all players' strategies, and $\mathbf{x}_{-i} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}, \mathbf{x}_{i+1}, \mathbf{x}_N\}$ the vector of all players' strategies except that of player i. We choose the average response time and average power consumption of a task as the performance metrics in the offloading game. The cost function of player i can be computed as follow

$$P_i(\mathbf{x}_i, \mathbf{x}_{-i}) = T_i + \mu_i E_i \tag{4}$$

where T_i is the average response time of all tasks generated by device i, E_i is the average energy consumption of the task generated by device i, and $\mu_i \in \mathbb{R}^+$ is the impact factor of energy consumption. The game with N devices is specified by $G = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N; P_1, P_2, \ldots, P_N\}$. The aim of device i, given other devices' strategies $\mathbf{x}_{-i}$, is to choose a strategy $\mathbf{x}_i \in X_i$ that minimizes his cost function $P_i(\mathbf{x}_i, \mathbf{x}_{-i})$ i.e., to

$$\min P_i(\mathbf{x}_i, \mathbf{x}_{-i}), \quad \text{subject to } \mathbf{x}_i \in X_i \tag{5}$$

It is a typical game problem. The objective function to be optimized is not only related to its strategy but also related to the strategies of other players.

B. THE PERFORMANCE METRIC

As described in the above part, the performance metric is the combination of average response time and average power consumption of a task. Tasks can be executed locally on devices or offloaded to satellites. We compute the average response time and average energy consumption in each case separately.

1) COMPUTATION TASK OFFLOADED TO SATELLITE

If the mobile device i offloads a task to satellite j, it takes three steps: offload a task, execute on satellite, and return the result. Due to the high dynamic of satellites, device i cannot always communicate with satellite j. Therefore, the waiting time for communication should be considered in both offloading and returning.

First, we compute the average response time of the task that is offloaded from device i to satellite j, which consists of five parts: average waiting time for offloading $T_{i,j}^{w\_off}$, average transmission time $T_{i,j}^{trans}$, average queue waiting time $T_{i,j}^{w\_que}$, average executing time $T_{i,j}^{sat}$, and average waiting time for returning $T_{i,j}^{w\_back}$. It is easy to get the expression of $T_{i,j}^{trans}$ and $T_{i,j}^{sat}$ as follow:

$$T_{i,j}^{trans} = \frac{\bar{d}}{R_{i,j}} \tag{6}$$

$$T_{i,j}^{sat} = \frac{\bar{c}}{C_j^{(s)}} \tag{7}$$

The waiting time caused by satellite orbiting is related to the percentage of time that device i can communicate with satellite j, i.e., θ_j. Based on the knowledge of probability, we can calculate $T_{i,j}^{w\_off}$ and $T_{i,j}^{w\_back}$ as below.

$$T_{i,j}^{w\_off} = \int_{\theta_j T_j}^{T_j} \frac{1}{T_j}\left(T_j - t\right) dt = \frac{T_j\left(1-\theta_j\right)^2}{2} \tag{8}$$

$$\begin{aligned} T_{i,j}^{w\_back} &= \int_{\theta_j T_j - T_{i,j}^{sat} - T_{i,j}^{w\_que}}^{\theta_j T_j} \frac{1}{\theta_j T_j}\left(T_i - T_{i,j}^{sat} - T_{i,j}^{w\_que} - t\right) dt \\ &= \frac{\left(1-\theta_j\right)}{\theta_j}\left(T_{i,j}^{sat} + T_{i,j}^{w\_que}\right) - \frac{\left(T_{i,j}^{sat} + T_{i,j}^{w\_que}\right)^2}{2\theta_j T_j} \end{aligned} \tag{9}$$

where T_j is the orbit period of satellite j, and $T_j \gg T_{i,j}^{sat} + T_{i,j}^{w\_que}$.

The process of executing tasks on satellite j is an M/G/1 queue system. Based on the queuing theory [34], the average queue waiting time of the tasks on satellite j is

$$T_{i,j}^{w\_que} = \frac{\widetilde{\lambda}_j}{2\left(1-\widetilde{\rho}_j\right)} \frac{\overline{c^2}}{\left(C_j^{(s)}\right)^2} \tag{10}$$

where $\widetilde{\lambda}_j = \sum_{i=1}^{N} x_{i,j}\lambda_i$ is the task arrival rate of satellite j, and $\widetilde{\rho}_j = \widetilde{\lambda}_j T_{i,j}^{sat}$ is the utilization of the queue on satellite j. Equation (10) can be simplified to

$$T_{i,j}^{w\_que} = \frac{\overline{c^2} \sum_{i=1}^{N} x_{i,j}\lambda_i}{2\left(C_j^{(s)} - \bar{c}\sum_{i=1}^{N} x_{i,j}\lambda_i\right) C_j^{(s)}} \tag{11}$$

Therefore, the average response time of the task that is offloaded from device i to satellite j is

$$\begin{aligned} T_{i,j} &= T_{i,j}^{w\_off} + T_{i,j}^{trans} + T_{i,j}^{w\_que} + T_{i,j}^{sat} + T_{i,j}^{w\_back} \\ &= \frac{T_j\left(1-\theta_j\right)^2}{2} + \frac{\bar{d}}{R_{i,j}} + \frac{\left(T_{i,j}^{sat} + T_{i,j}^{w\_que}\right)}{\theta_j} \\ &\quad - \frac{\left(T_{i,j}^{sat} + T_{i,j}^{w\_que}\right)^2}{2\theta_j T_j} \end{aligned} \tag{12}$$

Next, we compute the average energy consumption of the task that is offloaded from device i to satellite j, which consists of two parts, average transmission energy consumption $E_{i,j}^{trans}$ and average executing energy consumption $E_{i,j}^{sat}$.

According to the communication model, $E_{i,j}^{trans}$ can be computed as follow,

$$E_{i,j}^{trans} = p_i T_{i,j}^{trans} = \frac{p_i \bar{d}}{R_{i,j}} \tag{13}$$

The energy consumption is proportional to the square of the frequency of the CPU [30], [33]. Therefore, $E_{i,j}^{sat}$ can be computed as follow,

$$E_{i,j}^{sat} = \kappa \left(C_j^{(s)}\right)^2 \bar{c} \tag{14}$$

where κ is the effective switch capacitance that depends on the chip architecture [32].

The average energy consumption of the task offloaded from device i to satellite j is

$$E_{i,j} = E_{i,j}^{trans} + E_{i,j}^{sat} = \frac{p_i \bar{d}}{R_{i,j}} + \kappa \left(C_j^{(s)}\right)^2 \bar{c} \tag{15}$$

2) COMPUTATION TASK executed locally

Same as offloading to satellite, the process of executing tasks locally on device i is also an M/G/1 queue system. The average response time of the task executed on device i is equal to the average queue waiting time $T_{i,0}^{w\_que}$ plus the average executing time $T_{i,0}^{loc}$, i.e.,

$$T_{i,0} = T_{i,0}^{w\_que} + T_{i,0}^{loc} \tag{16}$$

Similarly, the local average executing time can be calculated as

$$T_{i,0}^{loc} = \frac{\bar{c}}{C_i^{(m)}} \tag{17}$$

Like (10), the average waiting time of the tasks that are executed on device i is

$$T_{i,0}^{wait} = \frac{\lambda_i}{2(1-\rho_i)} \frac{\overline{c^2}}{\left(C_i^{(m)}\right)^2} = \frac{x_{i,0}\lambda_i \overline{c^2}}{2\left(C_i^{(m)} - x_{i,0}\lambda_i \bar{c}\right) C_i^{(m)}} \tag{18}$$

where $\lambda_i = x_{i,0}\lambda_i$ is the task arrival rate of device i, and $\rho_i = \lambda_i T_{i,0}^{loc}$ is the utilization of the queue on device i.

Therefore, the average response time of the task that is executed locally on device i is

$$T_{i,0} = T_{i,0}^{w\_que} + T_{i,0}^{loc} = \frac{x_{i,0}\lambda_i \overline{c^2}}{2\left(C_i^{(m)} - x_{i,0}\lambda_i \bar{c}\right) C_i^{(m)}} + \frac{\bar{c}}{C_i^{(m)}}$$

Then we compute the average energy consumption of the task that is executed locally on device i. Like (14), the average energy consumption is

$$E_{i,0} = \kappa \left(C_i^{(m)}\right)^2 \bar{c} \tag{19}$$

Finally, the average response time of all tasks generated by device i is

$$T_i = x_{i,0}\left(T_{i,0}^{wait} + T_{i,0}^{loc}\right) + \sum_{j=1}^{M} x_{i,j} T_{i,j} \tag{20}$$

The average energy consumption of the task generated by device i is

$$E_i = x_{i,0} E_{i,0} + \sum_{j=1}^{M} x_{i,j} E_{i,j} \tag{21}$$

C. THE EXISTENCE AND UNIQUENESS OF NASH EQUILIBRIUM

In this part, we will analyze the existence and uniqueness of Nash equilibrium for the offloading game. First, two well-known lemmas [35], [36] are presented below.

Lemma 1: At least one Nash equilibrium for a non-cooperative game $G = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N; P_1, P_2, \ldots, P_N\}$ is existence if, for all $1 \le i \le$ N:

(1) The strategy space X_i is a non-empty, convex, and compact subset of some Euclidean space.

(2) The cost function P_i ($\mathbf{x}_i$, $\mathbf{x}_{-i}$) is continues and quasi-convex in X_i.

Lemma 2: A continuous and twice differentiable function $P(\mathbf{x})$, where $\mathbf{x} = (x_1, x_2, \ldots, x_M)$, is convex if and only if its Hessian matrix

$$\mathbf{H}(P(\mathbf{x})) = \left[\frac{\partial^2 P}{\partial x_i \partial x_j}\right]_{M \times M} \tag{22}$$

of second partial derivatives is positive semidefinite.

The following theorem gives the existence of the Nash equilibrium of the above game.

Theorem 1(Existence): There is a Nash equilibrium for the computation offloading game $G = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N; P_1, P_2, \ldots, P_N\}$, where P_i is defined by (4).

Proof: Obviously, X_i is a non-empty, convex, and compact subset. Next, we compute the Hessian matrix of P_i ($\mathbf{x}_i$, $\mathbf{x}_{-i}$). The partial derivative is

$$\frac{\partial P_i}{\partial x_{i,0}} = \frac{\lambda_i \overline{c^2}}{2C_i^{(m)}}\left(\frac{2x_{i,0}}{C_i^{(m)} - x_{i,0}\lambda_i\bar{c}} + \frac{\lambda_i \bar{c} x_{i,0}^2}{\left(C_i^{(m)} - x_{i,0}\lambda_i\bar{c}\right)^2}\right) + \frac{\bar{c}}{C_i^{(m)}} + \mu_i\kappa\left(C_i^{(m)}\right)^2\bar{c} \tag{23}$$

$$\frac{\partial P_i}{\partial x_{i,j}} = \frac{T_j(1-\theta_j)^2}{2} + \frac{\bar{d}}{R_{i,j}} + \frac{\bar{c}}{\theta_j C_j^{(s)}} + \frac{1}{\theta_j} T_{i,j}^{w\_que} + \frac{x_{i,j}}{\theta_j}\frac{\partial T_{i,j}^{w\_que}}{\partial x_{i,j}} - \frac{1}{2\theta_j T_j}\left(\frac{\bar{c}}{C_j^{(s)}} + T_{i,j}^{w\_que}\right)^2 - \frac{x_{i,j}}{\theta_j T_j}\left(\frac{\bar{c}}{C_j^{(s)}} + T_{i,j}^{w\_que}\right)\frac{\partial T_{i,j}^{w\_que}}{\partial x_{i,j}} + \mu_i\left(\frac{p_i\bar{d}}{R_{i,j}} + \kappa\left(C_j^{(s)}\right)^2\bar{c}\right) \tag{24}$$

for all $1 \le j \le M$. Here,

$$\frac{\partial T_{i,j}^{w\_que}}{\partial x_{i,j}} = \frac{\lambda_i \overline{c^2}}{2\left(C_j^{(s)} - \bar{c}\sum_{n=1}^{N} x_{n,j}\lambda_n\right)^2} \tag{25}$$

The second-order partial derivative is

$$\frac{\partial^2 P_i}{\partial x_{i,0}^2}=\lambda_i\overline{c^2}\left(\frac{1}{\left(C_i^{(m)}-x_{i,0}\lambda_i\bar{c}\right)^2}+\frac{2x_{i,0}}{\left(C_i^{(m)}-x_{i,0}\lambda_i\bar{c}\right)^3}\right) \tag{26}$$

$$\begin{aligned}\frac{\partial^2 P_i}{\partial x_{i,j}^2} &= \left(1-\frac{1}{T_j}\left(T_{i,j}^{sat}+T_{i,j}^{w\_que}\right)\right)\frac{x_{i,j}}{\theta_j}\frac{\partial^2 T_{i,j}^{w\_que}}{\partial x_{i,j}^2}\\ &+\frac{1}{\theta_j}\left(1-\frac{1}{T_j}\left(T_{i,j}^{sat}+T_{i,j}^{w\_que}\right)\right)\frac{\partial T_{i,j}^{w\_que}}{\partial x_{i,j}}\\ &+\frac{1}{\theta_j}\left(1-\frac{x_{i,j}}{T_j}\frac{\partial T_{i,j}^{w\_que}}{\partial x_{i,j}}\right)\frac{\partial T_{i,j}^{w\_que}}{\partial x_{i,j}}\end{aligned} \tag{27}$$

for all $1 \le j \le M$. Here,

$$\frac{\partial^2 T_{i,j}^{w\_que}}{\partial x_{i,j}^2}=\frac{\overline{c^2}\bar{c}\lambda_i^2}{\left(C_j^{(s)}-\bar{c}\sum_{n=1}^{N}x_{n,j}\lambda_n\right)^3} \tag{28}$$

From the above result, we can easily verify by straightforward algebraic manipulation that

$$\frac{\partial^2 P_i}{\partial x_{i,j}^2}>0 \tag{29}$$

for all $0 \le j \le M$, and

$$\frac{\partial^2 P_i}{\partial x_{i,j}\partial x_{i,k}}=0 \tag{30}$$

for all $0 \le j \ne k \le M$. Therefore, the Hessian matrix

$$\mathbf{H}\left(P_i\left(\mathbf{x}_i,\mathbf{x}_{-i}\right)\right)=\left[\frac{\partial^2 P_i}{\partial x_{i,j}\partial x_{i,k}}\right]_{(M+1)\times(M+1)} \tag{31}$$

is a diagonal matrix, and the elements on the main diagonal are positive. The Hessian matrix is positive definition. Thus, $P_i(\mathbf{x}_i, \mathbf{x}_{-i})$ is convex function of $\mathbf{x}_i$ for each fixed $\mathbf{x}_{-i}$, for all $1 \le i \le N$. Obviously, $P_i(\mathbf{x}_i, \mathbf{x}_{-i})$ is quasi-convex. By Lemma 1, there is at least one Nash equilibrium for the game $G = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N; P_1, P_2, \ldots, P_N\}$. Theorem 1 holds. We denote the Nash equilibrium is $\mathbf{x}^* = \{\mathbf{x}_1^*, \mathbf{x}_2^*, \ldots, \mathbf{x}_N^*\}$.

Moulin has proved in [37] that if a game is dominance-solvable (hence Cournot-stable), the game has a unique Nash equilibrium, which is said to be a Cournot-stable Nash equilibrium outcome. A sufficient condition for dominance-solvable in [37] is presented below.

Lemma 3: For a non-cooperative game $G = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N; P_1, P_2, \ldots, P_N\}$ whose Hessian matrix is positive definition, if we have:

$$\sum_{\substack{k=1,2,\ldots M\\ k\neq j}}\left|\frac{\partial^2 P_i}{\partial x_{i,j}\partial x_{i,k}}\right|<\left|\frac{\partial^2 P_i}{\partial x_{i,j}^2}\right|,\quad \forall i\in\mathbf{N} \tag{32}$$

Then G is dominance-solvable.

The following theorem gives the uniqueness of the Nash equilibrium of the above game.

Theorem 2 (Uniqueness): The computation offloading game $G = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N; P_1, P_2, \ldots, P_N\}$ has a uniqueness Nash equilibrium, where P_i is defined by (4).

Proof: From (29) and (30), it is easily verified

$$\sum_{\substack{k=1,2,\ldots M\\ k\neq j}}\left|\frac{\partial^2 P_i}{\partial x_{i,j}\partial x_{i,k}}\right|=0<\left|\frac{\partial^2 P_i}{\partial x_{i,j}^2}\right|,\quad \forall i\in\mathbf{N} \tag{33}$$

By Lemma 3, the game has a unique Nash equilibrium. Theorem 2 holds.

V. ALGORITHMS

In this section, an iterative algorithm is proposed to find the Nash equilibrium of the computation offloading strategy. The details are shown in Algorithm 1.

Algorithm 1 Search the Nash Equilibrium of Computation Offloading Strategy

Input: $N, M, \theta_j, T_j, \lambda_i, \bar{c}_i, \bar{d}_i, \overline{c_i^2}, \overline{d_i^2}, C_i^{(m)}, C_j^{(s)}, R_{i,j}, p_i, \kappa$, for all $1 \le i \le N$ and $1 \le j \le M$.
Output: $\mathbf{x}^* = \{\mathbf{x}_1^*, \mathbf{x}_2^*, \ldots, \mathbf{x}_N^*\}$: Nash equilibrium of offloading strategy.

```
1:  Initialize: x= {x_1, x_2,... x_N}
2:  t = 1
3:  While t < Max iterations do
4:    for i = 1 to N do
5:      find x'_i that minimizes the cost function P_i with x_{-i}
6:      compute the cost function P_i (x'_i) by (22)
7:      if P_i (x'_i) > P_i (x_i) then
8:        x'_i = x_i
9:      end if
10:   end for
11:   if ||x'_i - x_i|| > ε then
12:     t = t + 1
13:   else
14:     x* = x'_i
15:     return x*
16:   end if
17: end while
```

We set the initial strategy of each device as an even distribution, i.e., $\mathbf{x}_i = (1/(M+1), 1/(M+1), \ldots, 1/(M+1))$, for all $1 \le i \le N$. In each iteration, every device will search the best offloading strategy $\mathbf{x'}_i$ in the current situation and compute the minimum cost function $P_i(\mathbf{x'}_i)$. $\mathbf{x'}_i$ is solved by the Lagrange multiplier, i.e., solving the following equations.

$$\nabla L\left(\mathbf{x}_i,\phi\right)=0\Leftrightarrow\begin{cases}\dfrac{\partial P_i}{\partial x_{i,j}}=\phi, & j\in\mathbf{M}\\ \displaystyle\sum_{j=0}^{M}x_{i,j}=1\end{cases} \tag{34}$$

The derivative of the cost function has been derived in the proof process of Theorem 1. The classical bisection method is used to find the solution of (34). Let I denote the maximum length of iterations, and ε denote the accuracy requirement. Then, the time complexity is $O(M(\log(I/\varepsilon)))$.

According to the characteristic of the Nash Equilibrium, the strategy will only be updated when the cost function decreases. The algorithm terminates when the two strategies sets are close enough, i.e.,

$$\|\mathbf{x'}_i - \mathbf{x}_i\| = \sqrt{\sum_{i=1}^{N}\sum_{j=0}^{M}\left|x'_{i,j} - x_{i,j}\right|^2} < \varepsilon \tag{35}$$

The final converged strategy set $\mathbf{x}^*$ is the Nash equilibrium of the task offloading game, i.e., no device can reduce its cost function if all other devices adopt the strategies $\mathbf{x}^*_{-i}$. Since the optimal strategy for all devices needs to be solved, the overall time complexity of Algorithm 1 is $O(KMN(\log(I/\varepsilon)))$, where K is the number of iterations.

VI. SIMULATION

A. PARAMETER SETTING

In this section, we show illustrative results to demonstrate the performance of the proposed algorithm. In the simulation, the task generation rate $\lambda_i = 0.15 + 0.0075(i-1)$ tasks/second, the average number of computing resources $\bar{c}_i = 1{+}0.5(i{-}1)$ billion cycles, the second moments $\overline{c_i^2} = 1.6\bar{c}_i^2$, the average size of task $\bar{d}_i = 1{+}0.1(i-1)$ MB, the second moments $\overline{d_i^2} = 1.5\bar{d}_i^2$, the computation power of mobile device $C_i^{(m)} = 1{+}0.1(i{-}1)$ GHz, the computation power of satellite $C_j^{(s)} = 2.5 + 0.1(j-1)$ GHz, the uplink data rate $R_{i,j} = 10$MBps, effective switch capacitance $\kappa = 10^{-28}$. These parameters are obtained by referring to [11], [23]. We choose the Iridium constellation as the satellite system in the simulation. The specific parameters are shown in Table 2. The constellation consists of 66 satellites distributed over 6 orbital planes. The coordinate of the satellite is acquired by STK.

TABLE 2. The parameters of the satellite constellation.

Parameters	Value
Number of Planes	6
Number of Satellites per Planes	11
Semimajor Axis	7159.14km
Orbital period	100min
Inclination	86.4°

B. SIMULATION RESULTS

In Figure 4, using device 5 as an example, we show the convergence process of the offloading strategy. The number of devices is 10, the number of satellites is 22 (consisting of 2 planes), and the terminated error $\varepsilon = 10$. It is obvious that the strategy of device 5 changes with the increase of the iterations before getting the Nash equilibrium, and the strategy remains almost unchanged when gradually approaching the Nash equilibrium. Finally, the strategy is convergent after 63 iterations. It shows that the Nash equilibrium of the computation offloading game exists.

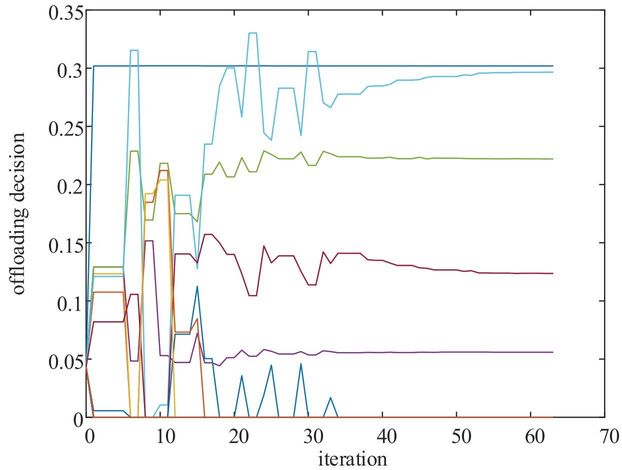


FIGURE 4. The strategy of device 5.

TABLE 3. The analysis of convergence rate.

Accuracy Requirement ε	Number of Iterations K
10^{-1}	11
$10^{-1.5}$	19
10^{-2}	28
$10^{-2.5}$	41
10^{-3}	57
$10^{-3.5}$	63
10^{-4}	72
$10^{-4.5}$	87
10^{-5}	93

Next, the convergence rate of the algorithm is discussed. The number of devices is 10, and the number of satellites is 11. In table 3, we show the number of iterations K for the accuracy requirement $\varepsilon = 10^{-1}, 10^{-1.5}, 10^{-2}, \ldots, 10^{-5}$. It seems that there is a rough linear growing trend of K with the increase of $\log(1/\varepsilon)$. The conclusion is consistent with the previous analysis of the time complexity of the algorithm.

Then, we do some research about the impact of some parameters on the performance of the algorithm. It reflects the adaptability of the proposed algorithm in satellite edge computing and helps the construction of satellite edge computing. Figure 5 shows the relationship between the average cost of device and the number of devices. The number of satellites is 22 (consisting of 2 planes). It is easily seen that the cost increases with the growth of the device number. It is because that the competition for resources among devices becomes more intense. Therefore, the average cost of a device is roughly positively related to the number of devices. Figure 5 also shows the performance of different strategies. The performance of the Nash equilibrium strategy solved by the proposed algorithm is much higher than other strategies. Particularly, when the number of devices is large, it has a more significant advantage. It is because when the competition is fierce, other strategies cannot balance the competition for resources among devices, resulting in a significant increase in cost. The proposed algorithm can effectively reduce the cost of device when the number of devices increases.

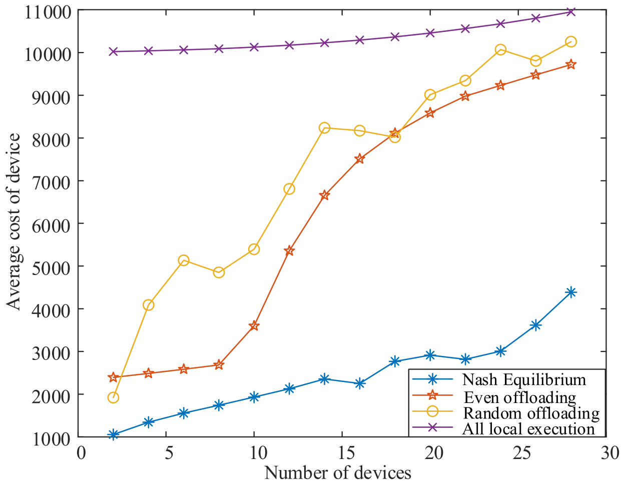


FIGURE 5. **The relationship between the average cost of a device and the number of devices under different strategies.**

Then, we discuss the impact of the number of satellites in two ways. One is the number of satellites per plane, and the other is the number of orbit planes. Figure 6 shows the relationship between the average cost of a device and the number of satellites per plane. The number of devices is 10, and the number of orbit plane is 1. The average cost of equipment is roughly negatively correlated with the number of satellites. As the number of satellites increases, the on-board resources increase, and the average cost decreases. The amount of cost reduction used by the proposed algorithm is higher than other algorithms. It suggests that the proposed algorithm can make more efficient use of resources.

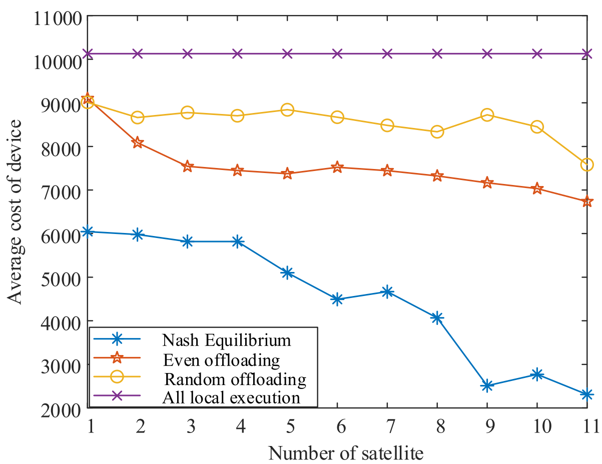


FIGURE 6. **The relationship between the average cost of a device and the number of satellites per plane under different strategies.**

Figure 7 shows the relationship between the average cost of a device and the number of orbit planes under Nash equilibrium strategy. When the number of satellites per orbital plane is constant, the average cost decrease as the number of orbital planes increases. However, as the number of orbit planes increases, the amount of the average cost reduction decreases. Additionally, if the number of satellites per plane is large, increasing the number of orbit plane does not effectively

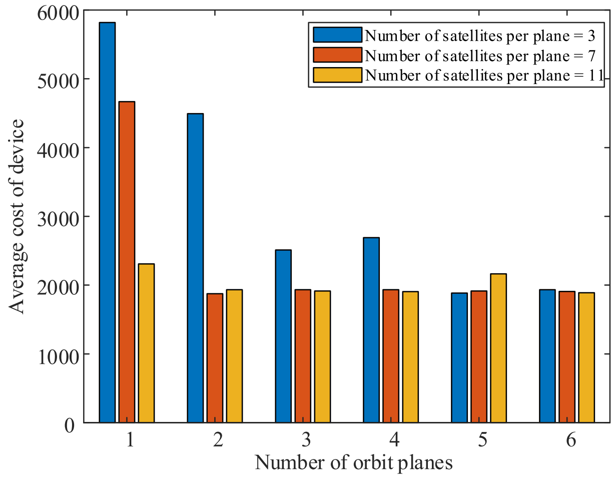


FIGURE 7. **The relationship between the average cost of a device and the number of orbit planes under Nash equilibrium strategy.**

reduce the cost. It is because when the on-board resources are saturated, increasing the number of satellites will not effectively reduce the cost of a device, but will lead to an increase in satellite system cost. Therefore, in order to achieve optimal system performance, satellite orbital parameters and quantities need to be optimized.

Figure 8 shows the relationship between the average cost of a device and the task generation rate of device under different strategies. The number of devices is 10, and the number of satellites is 22 (consisting of 2 planes). It is easily seen that the average cost of a device under Nash equilibrium will slowly increase with the growing of λ. When λ is small, the cost under the Nash equilibrium and all local execution are the same. It means that the offloading strategy converges to all tasks being executed locally. As the λ increases, if all the task is executed locally, the cost will increase dramatically until it is saturated. If the device uses an even offloading strategy, the cost is stable. For a random offloading strategy,

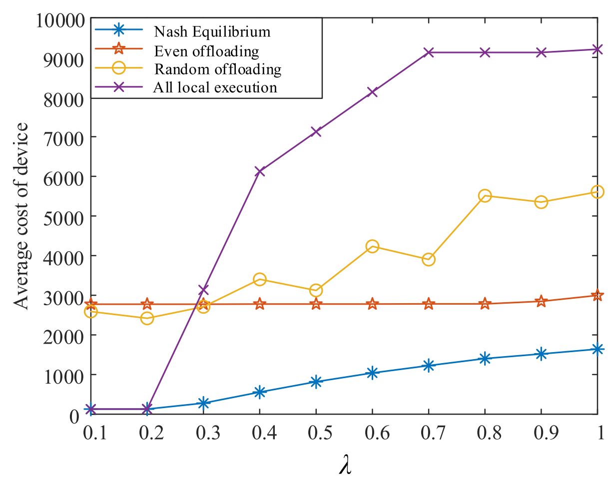


FIGURE 8. **The relationship between the average cost of a device and the task generation rate of device under different strategies.**

the cost is fluctuating, and the overall trend is on the rise. It shows that the proposed algorithm can effectively optimize the offloading strategy.

In general, the Nash Equilibrium strategy solved by the proposed algorithm can minimize the average cost of the device. It makes each device properly allocate tasks and utilize local and on-board resources under different parameters.

VII. CONCLUSION

In this paper, the game theory is used to optimize the computation offloading strategy of multiple mobile devices competing for on-board resources from multiple satellites in a satellite edge computing scenario. The system model of computation offloading is established in consideration of the intermittent communication caused by the high dynamic of satellites. We compute the average response time and average energy consumption of a task as the performance metrics. We establish a queuing model for multiple devices and multiple satellites and analytically obtain the game strategy and cost functions of the computation offloading game. The existence and uniqueness of the Nash equilibrium is theoretically proved, and an iterative algorithm is designed to find the Nash equilibrium strategy of each device. Finally, numerical simulations show that the effectiveness of the algorithm and the game-based offloading strategy can significantly reduce the average cost of device.

REFERENCES

[1] *Gartner Says 6.4 Billion Connected 'Things' Will Be in Use in 2016*, Gartner, Stamford, CA, USA, 2015. [Online]. Available: http://www.gartner.com/newsroom/id/3165317

[2] Z. Qu, G. Zhang, H. Cao, and J. Xie, "LEO satellite constellation for Internet of Things," *IEEE Access*, vol. 5, pp. 18391–18401, 2017.

[3] O. Kodheli, A. Guidotti, and A. Vanelli-Coralli, "Integration of satellites in 5G through leo constellations," in *Proc. IEEE Global Commun. Conf.*, Dec. 2017, pp. 1–6.

[4] L. Boero, R. Bruschi, F. Davoli, M. Marchese, and F. Patrone, "Satellite networking integration in the 5g ecosystem: Research trends and open challenges," *IEEE Netw.*, vol. 32, no. 5, pp. 9–15, Sep. 2018.

[5] H. El-Sayed, S. Sankar, M. Prasad, D. Puthal, A. Gupta, M. Mohanty, and C.-T. Lin, "Edge of Things: The big picture on the integration of edge, IoT and the cloud in a distributed computing environment," *IEEE Access*, vol. 6, pp. 1706–1717, 2018.

[6] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The cae for VM-based cloudlets in mobile computing," *IEEE Pervas. Comput.*, vol. 8, no. 4, pp. 14–23, Nov. 2009.

[7] K. Kumar and Y.-H. Lu, "Cloud computing for mobile users: Can offloading computation save energy?" *Computer*, vol. 43, no. 4, pp. 51–56, Apr. 2010.

[8] D. Sabella, A. Vaillant, P. Kuure, U. Rauschenbach, and F. Giust, "Mobile-edge computing architecture: The role of MEC in the Internet of Things," *IEEE Consum. Electron. Mag.*, vol. 5, no. 4, pp. 84–91, Oct. 2016.

[9] E. Ahmed and M. H. Rehmani, "Mobile edge computing: Opportunities, solutions, and challenges," *Future Gener. Comput. Syst.*, vol. 70, pp. 59–63, May 2017.

[10] G. Premsankar, M. Di Francesco, and T. Taleb, "Edge computing for the Internet of Things: A case study," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1275–1284, Apr. 2018.

[11] L. Yan, S. Cao, Y. Gong, H. Han, J. Wei, Y. Zhao, and S. Yang, "SatEC: A 5G satellite edge computing framework based on microservice architecture," *Sensors*, vol. 19, no. 4, p. 831, Feb. 2019.

[12] 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 Netw.*, vol. 33, no. 1, pp. 70–76, Jan. 2019.

[13] Y. Wang, J. Yang, X. Guo, and Z. Qu, "Satellite edge computing for the Internet of Things in aerospace," *Sensors*, vol. 19, no. 20, p. 4375, Oct. 2019.

[14] F. Wang, D. Jiang, S. Qi, C. Qiao, and J. Xiong, "Dynamic computing resource adjustment in edge computing satellite networks," in *Simulation Tools and Techniques* (Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering). Springer, 2019, pp. 135–145.

[15] J. Wei, J. Han, and S. Cao, "Satellite IoT edge intelligent computing: A research on architecture," *Electronics*, vol. 8, no. 11, p. 1247, Oct. 2019.

[16] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1628–1656, Mar. 2017.

[17] S. Kaur and K. Kaur, "A survey on computation offloading techniques in mobile cloud computing and their parametric comparison," in*Innovations in Computer Science and Engineering* (Lecture Notes in Networks and Systems). Singapore: Springer, 2017, pp. 81–87.

[18] J. Wang, J. Pan, F. Esposito, P. Calyam, Z. Yang, and P. Mohapatra, "Edge cloud offloading algorithms: Issues, methods, and perspectives," *Comput. Surv.*, vol. 52, no. 1, pp. 1–23, Feb. 2019.

[19] J. Wang, J. Peng, Y. Wei, D. Liu, and J. Fu, "Adaptive application offloading decision and transmission scheduling for mobile cloud computing," *China Commun.*, vol. 14, no. 3, pp. 169–181, Mar. 2017.

[20] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, "Delay-optimal computation task scheduling for mobile-edge computing systems," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2016, pp. 1451–1455.

[21] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3590–3605, Dec. 2016.

[22] K. Zhang, Y. Mao, S. Leng, Q. Zhao, L. Li, X. Peng, L. Pan, S. Maharjan, and Y. Zhang, "Energy-efficient offloading for mobile edge computing in 5G heterogeneous network," *IEEE Access*, vol. 4, pp. 5896–5907, 2016.

[23] K. Li, "A game theoretic approach to computation offloading strategy optimization for non-cooperative users in mobile edge computing," *IEEE Trans. Sustain. Comput.*, to be published.

[24] V. Cardellini, V. De Nitto Personé, V. Di Valerio, F. Facchinei, V. Grassi, F. Lo Presti, and V. Piccialli, "A game-theoretic approach to computation offloading in mobile cloud computing," *Math. Program.*, vol. 157, no. 2, pp. 421–449, Jun. 2016.

[25] H. Cao and J. Cai, "Distributed multiuser computation offloading for cloudlet-based mobile cloud computing: A game-theoretic machine learning approach," *IEEE Trans. Veh. Technol.*, vol. 67, no. 1, pp. 752–764, Jan. 2018.

[26] J. Zheng, Y. Cai, Y. Wu, and X. Shen, "Dynamic computation offloading for mobile cloud computing: a stochastic game-theoretic approach," *IEEE Trans. Mobile Comput.*, vol. 18, no. 4, pp. 771–786, Apr. 2019.

[27] L. Liu, X. Guo, Z. Chang, and T. Ristaniemi, "Joint optimization of energy and delay for computation offloading in cloudlet-assisted mobile cloud computing," *Wireless Netw.*, vol. 25, no. 4, pp. 2027–2040, May 2019.

[28] W. Labidi, M. Sarkiss, and M. Kamoun, "Energy-optimal resource scheduling and computation offloading in small cell networks," in *Proc. 22nd Int. Conf. Telecommun. (ICT)*, Apr. 2015, pp. 313–318.

[29] B. Elbert, *Introduction to Satellite Communication*. Norwood, MA, USA: Artech House, 2008.

[30] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 4, pp. 974–983, Apr. 2015.

[31] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2795–2808, Oct. 2016.

[32] T. D. Burd and R. W. Brodersen, "Processor design for portable systems," *Technol. Wireless Comput.*, vol. 3, pp. 119–137, 1996.

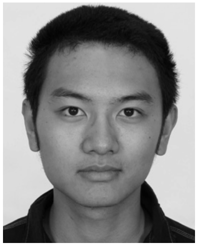
[33] H. Li, "Multi-task offloading and resource allocation for energy- efficiency in mobile edge computing," *Int. J. Comput. Techn.*, vol. 5, no. 1, pp. 5–13, 2018.

[34] L. Kleinrock, *Queueing Systems*, vol. 1. Hoboken, NJ, USA: Wiley, 1975.

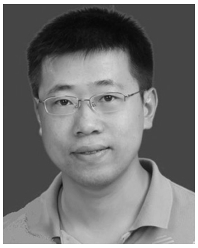
[35] A. B. MacKenzie and L. A. DaSilva, "Game theory for wireless engineers," *Synthesis Lectures Commun.*, vol. 1, no. 1, pp. 1–86, 2006.

[36] A. W. Roberts, "Convex functions," in *Handbook Convex Geometry*, vol. 1993. Amsterdam, The Netherlands: Elsevier, pp. 1081–1104.

[37] H. Moulin, "Dominance solvability and cournot stability," *Math. Social Sci.*, vol. 7, no. 1, pp. 83–102, Feb. 1984.



YUXUAN WANG received the B.S. and M.S. degrees in instrument science and technology from the National University of Defense Technology, in 2013 and 2015, respectively, where he is currently pursuing the Ph.D. degree in instrument science and technology. His research interests include onboard satellite on-board software, satellite function modeling, and software development environment.



XIYE GUO received the B.S., M.S., and Ph.D. degrees from the College of Mechatronics Engineering and Automation, National University of Defense Technology, China, in 2003, 2005, and 2010, respectively. From December 2016 to February 2017, he was a Visiting Scholar with the University of New South Wales (UNSW). He is currently an Associate Research Fellow with the College of Mechatronics Engineering and Automation, National University of Defense Technology. His current research interests include GNSS, navigation augmentation, and precise positioning and timing.

JUN YANG received the B.S., M.S., and Ph.D. degrees from the College of Mechatronics Engineering and Automation, National University of Defense Technology, China, in 1994, 1999, and 2007, respectively. He is currently a Professor with the College of Intelligence Science and Technology, National University of Defense Technology. He is responsible for teaching and research in signal processing, space instruments, GNSS applications, and intelligent satellite.

ZHI QU received the B.S., M.S., and Ph.D. degrees in instrument science and technology from the National University of Defense Technology, China, in 2008, 2010, and 2016, respectively. His research interests include digital signal processing, satellite tracking telemetry, command (TT&C), spread spectrum communication, and intelligent satellite.