Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 4212548, 12 pages
https://doi.org/10.1155/2021/4212548

Research Article

Collaborative Computing and Resource Allocation for LEO Satellite-Assisted Internet of Things

Tao Leng,[1,2] Xiaoyao Li,[1,2] Dongwei Hu,[3] Gaofeng Cui,[1,2,3] and Weidong Wang[1,2]

[1]*School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2]*Key Laboratory of Universal Wireless Communications, Ministry of Education, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[3]*Science and Technology on Information Transmission and Dissemination in Communication Networks Laboratory, 54th Research Institute of China Electronics Technology Group Corporation, Shijiazhuang, China*

Correspondence should be addressed to Gaofeng Cui; cuigaofeng@bupt.edu.cn

Received 13 May 2021; Revised 5 August 2021; Accepted 27 August 2021; Published 15 September 2021

Academic Editor: Miaowen Wen

Satellite-assisted internet of things (S-IoT), especially the S-IoT based on low earth orbit (LEO) satellite, plays an important role in future wireless systems. However, the limited on-board communication and computing resource and high mobility of LEO satellites make it hard to provide satisfied service for IoT users. To maximize the task completion rate under latency constraints, collaborative computing and resource allocation among LEO networks are jointly investigated in this paper, and the joint task offloading, scheduling, and resource allocation is formulated as a dynamic mixed-integer problem. To tack the complex problem, we decouple it into two subproblems with low complexity. First, the max-min fairness is adopted to minimize the maximum latency via optimal resource allocation with fixed task assignment. Then, the joint task offloading and scheduling is formulated as a Markov decision process with optimal communication and computing resource allocation, and deep reinforcement learning is utilized to obtain long-term benefits. Simulation results show that the proposed scheme has superior performance compared with other referred schemes.

1. Introduction

Internet of things (IoT) plays an important role in future intelligent society, and many techniques have been evaluated and implemented to provide better service for the data transmission in IoT network. Although the fifth generation (5G) wireless systems can support massive machine type communication (mMTC), it mainly focuses on the terrestrial network-based IoT. For depopulated areas where lack telecommunication infrastructures, satellite communication has been adopted as an important component for 5G beyond or the sixth generation (6G) wireless systems [1, 2]. Moreover, edge computing refers to the techniques that shift the computing units to the access nodes near the user equipment, or the data is processed at the user equipment locally [3]. With edge computing, the access delay can be reduced, and the radio resource can also be utilized efficiently. Benefit from the development of satellite on-board processing techniques [4], edge computing-enhanced satellite networks have also become a hot topic for integrated satellite and terrestrial networks [2, 5, 6]. Thus, edge computing-enhanced satellite-assisted IoT (S-IoT) receives lots of attention in both industrial and academic areas [7, 8].

For edge computing-enhanced S-IoT networks, the IoT devices and satellites are both resource-limited. Therefore, the joint computing and communication resource allocation for tasks generated by users (IoT devices) is important for improving the performance of the systems [9]. Moreover, the characteristics of the satellite networks will also affect the mechanisms used for the resource management [10]. Generally, the existing satellite communication systems can be divided into three categories, which are geosynchronous earth orbit (GEO), medium earth orbit (MEO), and low earth orbit (LEO). Among the three categories of satellite systems, LEO satellites with the lowest propagation delay are emerging as an important component for future integrated satellite and terrestrial networks [11–13]. In this paper, we also focus on the edge computing-enhanced LEO satellite networks, and the advantages of LEO satellite-assisted IoT over the other satellite systems can be summarized as follows:

(1) The propagation delay introduced by the LEO satellite is low. For example, the one-trip propagation delay is about 5 ms for LEO satellites located at the height of 1500 km. While it is about 120 ms for GEO satellites

(2) Though the satellite on-board processing capability is usually limited, multiple satellites in the LEO networks, especially the mega-constellation LEO networks, can form a virtual resource pool, which can be utilized to improve the performance of the edge computing enhanced S-IoT

(3) Multiple LEO satellites can generate overlapped coverage areas, and the communication and computing resource can be allocated flexibly

From the above analysis, the LEO S-IoT can benefit from the low propagation delay and collaborative processing among multiple satellites. However, the edge computing-enhanced LEO S-IoT still faces several problems. First, the varying topology of LEO networks makes it difficult to manage the limited communication and computing resource dynamically [14]. Second, the resource management should jointly consider multiple types of links, such as satellite-to-ground and satellite-to-satellite. Last but not the least, the communication and computing resource should be jointly allocated.

In this paper, edge computing-enhanced LEO S-IoT is considered, and the task generated by the users needs to be handled locally or via the LEO networks collaboratively. Different from the existing studies, collaborative computing among multiple satellites is utilized to reduce the on-board processing latency. Moreover, the satellite-to-ground and satellite-to-satellite links are considered jointly for communication and computing resource allocation. The main contributions are listed as below.

(i) A framework for collaborative computing among multiple LEO satellites with varying topology is provided, and the effects of satellite-to-ground and satellite-to-satellite links on the processing latency are jointly considered

(ii) The collaborative computing and resource allocation for user tasks are formulated as a joint task offloading, scheduling, and multidimensional resource allocation problem to maximize the completion rate of tasks, and the complex problem is divided into two subproblems with low complexity

(iii) Deep reinforcement learning (DRL) and max-min fairness optimization are adopted to achieve long-term benefits in terms of task completion rate, and simulation results verify the performance of the proposed algorithms

The remainder of this paper is organized as follows. Section 2 summarizes the related works. The system model and problem formulation are described in Section 3. In Section 4, resource allocation based on max-min fairness and task scheduling and offloading with DRL is analyzed, respectively. Section 5 evaluates the proposed algorithms, and Section 6 concludes this paper.

2. Related Works

Joint task offloading, scheduling, and resource allocation plays an important role in edge computing enhanced S-IoT networks. Papa et al. evaluate the reconfigurable software-defined network with LEO constellation and propose an optimal controller placement and satellite-to-controller assignment method which can minimize the average flow setup time [15]. Liu et al. propose a task-orient network architecture for edge computing enhanced space-air-ground-aqua integrated networks [9]. Xie et al. analyze the joint caching, communication, and computing resource management for space information networks [2]. Although the joint task offloading, scheduling, and resource allocation for edge computing enhanced S-IoT is highlighted in the existing works [2, 9, 15], the resource management methods are not given in detail.

Cheng et al. propose a computing offloading method for IoT applications in space-air-ground integrated network with fixed data rate [16]. Cao et al. propose an edge-cloud architecture based on software-defined networking and network function virtualization for the space-air-ground integrated network [17]. Wang et al. introduce the hardware and software structure for the edge computing-enhanced S-IoT [18]. A fine-grained resource management scheme is introduced by Wang et al. for edge computing-enhanced satellite networks [19]. Yan et al. propose a 5G satellite edge computing framework based on microservice architecture with the embedded hardware platform [5]. LiWang et al. investigate the computing offloading methods with delay and cost constraints for satellite-ground internet of vehicles [20]. Jiao et al. analyze a joint network stability and resource allocation optimization problem for high-throughput satellite-based IoT [21]. An orbital edge computing architecture is introduced by Denby and Lucia, and the power and software optimization for the orbital edge are also analyzed [22]. Cui et al. propose a joint offloading and resource allocation for GEO satellite-assisted vehicle-to-vehicle communication [23]. A collaborative computing and resource allocation method among multiple user pairs is given by Zhang et al. for GEO S-IoT [24]. Song et al. propose a mobile edge computing framework for terrestrial-satellite IoT, and an energy-efficient computing offloading and resource allocation method is used to minimize the weighted sum energy consumption [25]. A learning-based queue-aware task offloading and resource allocation algorithm is analyzed by Liao et al. for space-air-ground-integrated power IoT [26]. Tang et al. present a hybrid cloud and edge computing LEO satellite network and investigate the computation offloading decisions to minimize the sum energy consumption of ground users [27].

Though some exiting works listed above investigate the task offloading and resource allocation for edge computing-enhanced S-IoT, none of these works consider the collaborative computing among multiple LEO satellites. Moreover, the joint optimization of satellite-to-ground and satellite-to-satellite links is not investigated either.

3. System Model and Problem Formulation

In this paper, we consider a typical scenario, as shown in Figure 1, consisting of multiple terrestrial users and a LEO satellite constellation. The LEO satellite constellation is deployed with intersatellite links (ISLs) for cooperative processing among satellites, and each satellite can exchange information with four adjacent satellites through ISLs. With the network topology described above, the tasks generated by the users arrive randomly as a time series, and the tasks can be processed locally or offloaded to satellites for processing. Although the computing resource of a single satellite is scarce due to the characteristics of the on-board devices, computing units on multiple satellites can form a collaborative computing pool, and the satellites which are overloaded can forward the tasks that need to be handled to the other satellites with light load. Thus, the task offloaded to the satellites can be handled by its serving satellite or other satellites available via ISLs. Moreover, the computing resource available for the IoT devices is also limited, and tasks that need to be processed locally should be handled one by one. While for satellites, tasks from multiple users can be handled in parallel, and the resource is shared among tasks belong to multiple users.

Moreover, every satellite needs to maintain the satellite-to-ground transmission queue and the on-board processing queue. For satellite-to-ground transmission, the tasks offloaded to satellites will be scheduled slot by slot, and the tasks cannot be partitioned. When the task arrives at the satellite used for data processing, it will enter the processing queue and wait for data processing. To guarantee the efficiency and reliability of transmission and data processing, the communication/computing resource allocated to user tasks will be occupied until the end of the transmission/processing. After the data processing, the results will be delivered to the users. During the task offloading processes, the resource occupancy and mobility of the LEO satellites will both affect the performance of the system. For tasks handled locally, the latency is mainly composed of waiting latency in queue and processing latency. While for tasks handled by satellites, several factors, which are transmission latency, propagation delay, and processing latency, need to be considered. Thus, the system models adopted and problem formulation are listed in the following subsections.

3.1. *Satellite Orbit Model.*

In earth-centered inertial (ECI) coordinate system, the position of the satellite in space can be described by orbital elements, namely, eccentricity e, semimajor axis a, inclination i, right ascension of the ascending node (RAAN) Ω, argument of periapsis ω, and initial true anomaly φ. In this paper, we consider the satellite orbit to be a circular orbit with $e = 0$ and $\omega = 0$, so the ECI coordinate of satellite at time t can be expressed as

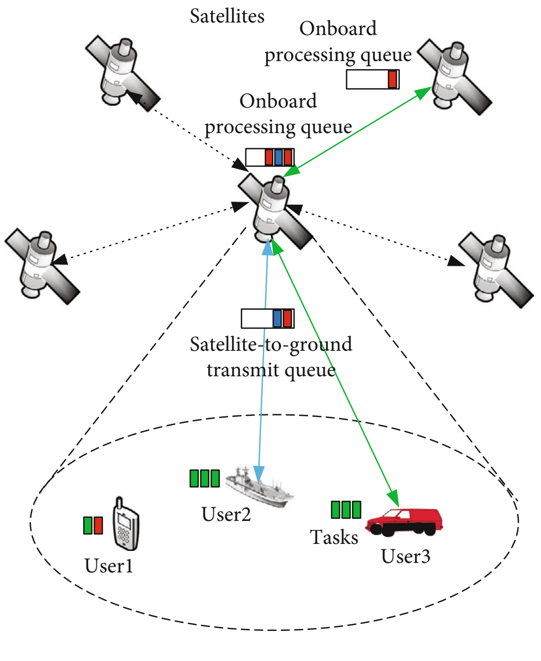

FIGURE 1: Collaborative computing among multiple LEO satellites.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{\mathrm{ECI}} = (R+h) \begin{bmatrix} \cos\Omega\, \cos\left(\varphi'(t)\right) - \sin\Omega\, \sin\left(\varphi'(t)\right) \cos i \\ \sin\Omega\, \cos\left(\varphi'(t)\right) + \cos\Omega\, \sin\left(\varphi'(t)\right) \cos i \\ \sin\Omega\, \sin\left(\varphi'(t)\right) \sin i \end{bmatrix}, \quad (1)$$

where R is the earth radius, h is the height of satellite, and $a = R + h/2$. $\varphi'(t) = \varphi + \omega_s t$, where $\omega_s = \sqrt{GM_e/(R+h)^3}$ denotes the angular velocity of satellite, which is related to the altitude of satellite, gravitational constant G, and mass of the earth M_e. $t = \rho(l-1)$ denotes the running time of the satellite at the beginning of time slot l, in which ρ is the length of one time slot.

To obtain the coordinates of satellite n, the RAAN Ω_n and initial true anomaly φ_n also needs to be calculated. Since the Walker constellation is symmetrical and all satellites adopt circular orbits with the same height and the same inclination, the orbital plane is evenly distributed along the equator, and the satellites are evenly distributed in the orbital plane. The phase relationship of the satellites in different orbital planes can be expressed as

$$\begin{cases} \Omega_n = \dfrac{2\pi}{P}(P_n - 1), \\ \varphi_n = \dfrac{2\pi}{S}(N_n - 1) + \dfrac{2\pi}{N}F(P_n - 1), \end{cases} \quad (2)$$

where N denotes the number of satellites, P denotes the number of orbits, S denotes the number of satellites in each orbit, thus, $N = P \times S$. P_n is the serial number of the orbit where the satellite n is located, and $P_n = \lfloor n/S \rfloor + 1$. N_n is the serial number of satellite n in its orbit, and $N_n = n - (P_n - 1)S$. F denotes the phase factor of orbit. With (1) and (2), we can obtain the ECI coordinate of any satellite in the constellation at any time slot.

3.2. Coverage Model. Generally, the users' coordinates are expressed with longitude, latitude, and altitude (LLA) in the geographic coordinate system. To derive the situation of coverage of satellite at time slot l, we first need to convert the ECI coordinate of the satellite to earth centered earth fixed (ECEF) coordinate with the following formula

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{\mathrm{ECEF}} = \begin{bmatrix} \cos \eta_g & -\sin \eta_g & 0 \\ \sin \eta_g & \cos \eta_g & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{\mathrm{ECI}}, \quad (3)$$

where $\eta_g = \eta_{g_0} + \omega_e t$, in which η_{g_0} denotes the Greenwich hour angle at the beginning of the first time slot, and ω_e denotes the angular velocity of the earth's rotation. Then, we need to convert the ECEF coordinate of the satellite to LLA coordinate (lon, lat, alt) according to

$$\begin{cases} lon = \arctan \dfrac{y}{x}, \\ lat = \arctan \dfrac{z + Je^2 \sin lat}{\sqrt{x^2 + y^2}}, \\ alt = \dfrac{z}{\sin lat} - J\left(1 - e^2\right), \end{cases} \quad (4)$$

where $J = a/1 - e^2 \sin^2 lat$. Since $e = 0$, $J = a$ can be achieved. With (3) and (4), we can obtain the LLA coordinate of users and satellites, and the elevation angle σ of user can be expressed as

$$\sigma = \arctan \frac{\cos \Delta lon \cos lat_u \cos lat_s + \sin lat_u \sin lat_s - R/R + h}{\sqrt{1 - (\cos \Delta lon \cos lat_u \cos lat_s + \sin lat_u \sin lat_s)^2}}, \quad (5)$$

where $\Delta lon = lon_u - lon_s$, lon_u and lat_u denote user's longitude and latitude, respectively, and lon_s and lat_s denote satellite's longitude and latitude, respectively. We consider user m is covered by satellite n when σ_n^m is larger than the minimum elevation angle $\sigma_{\min}$. At the beginning, user m will select the satellite n_a with the smallest elevation angle for association, and when σ_n^m is less than $\sigma_{\min}$ as the associated satellite moves, the user will select the satellite n_h according to the elevation angle at the current time slot for handover.

3.3. Channel Model. In the scenario shown in Figure 1, two kinds of links should be considered, which are satellite-to-ground (StG) and satellite-to-satellite (StS) links. For StG links, line-of-sight (LOS) channel is assumed to be always existing between satellites and users on ground. Since we focus on the resource management for the satellite systems, only the path loss affected by the distance between satellites and users is considered in this paper, and the free space path loss (FSPL) model is adopted. Moreover, the channel quality of the users on ground, which are indicated by signal-to-noise ratio (SNR), will be quantified and transmitted to the satellites via the control channels. For StS links, point-to-point optical links are assumed to be implemented, and the channel capacity of the StS links is assumed to be large enough for the data transmission between satellites.

3.4. Task Arrival Model. Generally, LEO satellite-assisted IoT is suitable for multiple kinds of services, such as object identification and tracking and assets monitoring. In this paper, the tasks of each user are assumed to arrive continuously, and the task arrival follows Poisson distribution. The probability of κ tasks arriving in l time slots can be expressed as

$$P(T(l) = \kappa) = \frac{(\lambda l)^\kappa e^{-\lambda l}}{\kappa!}, \quad (6)$$

where λ denotes the rate of task arrival. And the time interval of task arrival follows the exponential distribution with parameter λ. Moreover, multiple tasks belong to one single user can only be handled with a first-in-first-out policy, and a single task cannot be scheduled until the processing results of its previous task are sent back to the user. More complicated scenarios, where multiple tasks of a single user are scheduled simultaneously, will be considered in future work.

3.5. Latency Model. Since the task can be handled locally or offloaded to satellites, the latency τ_k^m of task k generated by user m will be analyzed with three possible cases.

If task k is handled locally by user m, the latency τ_k^m can be expressed as

$$\tau_k^m = \tau_{k,m}^{\mathrm{wait}} + \tau_{k,m}^{\mathrm{process}}, \quad (7)$$

where $\tau_{k,m}^{\mathrm{wait}}$ denotes the waiting latency in queue and waiting latency due to the local resources being occupied by the task being processed. $\tau_{k,m}^{\mathrm{process}} = T_k^m / X_m$ denotes the processing latency, T_k^m is the number of bits in task k of user m, and X_m is the local computing resource of user m.

If task k is handled by satellite n_a associated with user m, the latency τ_k^m can be expressed as

$$\tau_k^m = \tau_{k,\mathrm{m}}^{\mathrm{off}} + \tau_{k,m}^{\mathrm{process}} + \tau_{k,m}^{\mathrm{return}}, \quad (8)$$

where $\tau_{k,m}^{\mathrm{off}}$ is the latency of offloading task k from user m to its associated satellite n_a, and it consists of waiting latency $\tau_{k,m}^{w\_\mathrm{trans}}$ for transmission, transmission latency $\tau_{k,m}^{\mathrm{trans}}$, and propagation latency $\tau_{k,m}^{\mathrm{prop}}$. Moreover, $\tau_{k,m}^{\mathrm{trans}} = T_k^m / C_{n_a}^m$, where $C_{n_a}^m$ denotes the data rate allocated by satellite n_a to task k of user m, and $C_{n_a}^m$ is affected by the available communication resource, such as power, bandwidth. $\tau_{k,m}^{\mathrm{prop}} = d_{m,n_a}/c$, where c is the speed of light, and d_{m,n_a} is the distance from user m to satellite n_a. $\tau_{k,m}^{\text{process}}$ composed of waiting latency in processing queue $\tau_{k,m}^{w\_\text{pro}}$ and processing latency $\tau_{k,m}^{\text{pro}}$, and $\tau_{k,m}^{\text{pro}} = T_k^m / X_{n_a}^m$, where $X_{n_a}^m$ denotes the computing resource allocated to task k of user m by satellite n_a. $\tau_{k,m}^{\text{return}}$ is the latency caused by sending the results back to user. Note that the transmission latency of return link is omitted, because the number of bits of computing results is usually very small. If the result can be returned to user within the coverage time of satellite n_a, $\tau_{k,m}^{\text{return}} = d_{n_a,m}/c$, otherwise, $\tau_{k,m}^{\text{return}} = d_{n_a,n_h} + d_{n_h,m}/c$, and d_{n_a,n_h} denotes the routing distance from satellite n_a to satellite n_h, and $d_{n_h,m}$ denotes the routing distance from satellite n_h to user m.

If task k is offloaded to satellite n_p through satellite n_a for processing, the latency τ_k^m can be expressed as

$$\tau_k^m = \tau_{k,m}^{\text{off}} + \tau_{k,m}^{\text{ISL}} + \tau_{k,m}^{\text{process}} + \tau_{k,m}^{\text{return}}, \tag{9}$$

where $\tau_{k,m}^{\text{off}}$ is the latency of offloading task k from user m to its associated satellite n_a. $\tau_{k,m}^{ISL}$ is the propagation latency of task k routing from satellite n_a to satellite n_p through ISLs. Suppose that optical links are adopted for intersatellite communication, and the ISLs data rate is high enough, the transmission latency of ISLs can be omitted. Therefore, $\tau_{k,m}^{\text{ISL}} = d_{n_a,n_p}/c$, in which d_{n_a,n_p} denotes the routing distance from satellite n_a to satellite n_p, and the routing strategy based on minimum distance is adopted. $\tau_{k,m}^{\text{process}}$ is composed of waiting latency $\tau_{k,m}^{w\_\text{pro}}$ in processing queue and processing latency $\tau_{k,m}^{\text{pro}}$ on satellite n_p. $\tau_{k,m}^{\text{pro}} = T_k^m / X_{n_p}^m$, where $X_{n_p}^m$ denotes the computing resource allocated to task k of user m by satellite n_p. $\tau_{k,m}^{\text{return}}$ is the propagation latency of returning the result back to user. First, the result will be routed from satellite n_p to satellite n_h, which is the serving satellite of user m. And then, the result will be sent by satellite n_h to user m. Thus, $\tau_{k,m}^{\text{return}} = d_{n_p,n_h} + d_{n_h,m}/c$.

3.6. Problem Formulation. In this paper, we intend to achieve the collaborative computing among multiple satellites through ISLs and maximize the completion rate of user tasks under latency constraints. Thus, the problem can be formulated as

$$\begin{aligned} \max_{\bar{\Omega}_l, \bar{\mathbf{\Psi}}_l, \bar{\text{O}}_l, \text{C}_l, \text{X}_l} \quad & \frac{\sum_l \sum_m \sum_k v_k^m}{\sum_l \sum_m \sum_k w_k^m} \\ s.t. \quad & \sum_{m \in \bar{\mathbf{\Phi}}_{n,l}} C_{n,l}^m \le \bar{C}_{n,l}, \\ & \sum_{m \in \bar{\mathbf{\Theta}}_{n,l}} X_{n,l}^m \le \bar{X}_{n,l}, \end{aligned} \tag{10}$$

where $w_k^m = 1$ when new task k arrives at user m, and $v_k^m = 1$ when $\tau_k^m \le \tau_{\max}$, otherwise, $v_k^m = 0$. $\tau_{\max}$ is the maximum latency constraint for the task k of user m. $\bar{\Omega}_l = \{\bar{\Omega}_{1,l}, \bar{\Omega}_{2,l}, \cdots, \bar{\Omega}_{M,l}\}$ and $\bar{\Omega}_{m,l} \in \{0, 1\}$, $\bar{\Omega}_{m,l} = 1$ denotes that the task of user m will be handled locally at time slot l. $\bar{\mathbf{\Psi}}_l = \{\bar{\Psi}_{1,l}, \bar{\Psi}_{2,l}, \cdots, \bar{\Psi}_{M,l}\}$ and $\bar{\Psi}_{m,l} \in \{1, 2, \cdots, N\}$, $\bar{\Psi}_{m,l} = n$ denotes that the task k of user m will be offloaded to satellite n at time slot l for processing. $\bar{\text{O}}_l = \{\bar{\text{O}}_{1,l}, \bar{\text{O}}_{2,l}, \cdots, \bar{\text{O}}_{\text{M},l}\}$ and $\bar{\text{O}}_{\text{m},l} \in \{0, 1\}$, $\bar{\text{O}}_{\text{m},l} = 1$ denotes that the task k of user m will be scheduled to be transmitted or processed at time slot l. $\text{C}_l = \{C_{1,l}, C_{2,l}, \cdots, C_{N,l}\}$ and $C_{n,l} = \{C_{n,l}^1, C_{n,l}^2, \cdots, C_{n,l}^M\}$ denote the communication resource allocated by satellite n to task k of user m at time slot l. $\text{X}_l = \{X_{1,l}, X_{2,l}, \cdots, X_{N,l}\}$ and $X_{n,l} = \{X_{n,l}^1, X_{n,l}^2, \cdots, X_{n,l}^M\}$ denote the computing resource allocated by satellite n to task k of user m at time slot l. $\bar{C}_{n,l}$ and $\bar{X}_{n,l}$ are the available communication resource and computing resource of satellite n at time slot l, respectively. $\bar{\mathbf{\Phi}}_{n,l}$ and $\bar{\mathbf{\Theta}}_{n,l}$ denote the tasks scheduled in transmission queue and processing queue of satellite n at time slot l, respectively.

From (10), we can see that the completion rate of tasks is affected by the offloading decision, scheduling decision, and resource allocation at each time slot. In addition, the offloading decision, scheduling decision, and resource allocation at the current time slot l will affect the states of time slot $l+1$. For instance, if the task cannot complete the transmission from user to satellite at time slot l, it cannot be processed at time slot $l+1$, and the communication resource occupied cannot be released to other tasks. Thus, the problem formulated in (10) can be seen as a dynamic programming problem based on joint offloading decision, scheduling decision, and resource allocation, which is difficult to be solved with traditional methods. To tackle the problem, we will decompose the complex problem into two subproblems.

4. Task Completion Rate Optimization Based on Deep Reinforcement Learning

According to Section 3, task offloading and scheduling decision indicators are discrete, while the resource allocation variables are continuous. Therefore, the problem formulated in (10) is a dynamic mixed-integer problem, which is nonconvex and difficult to find the optimal solutions. To solve this problem, we decompose the problem into two subproblems to reduce its complexity. The first subproblem is communication and computing resource allocation with fixed offloading decision and scheduling decision, which will be solved based on max-min fairness. The second subproblem is the joint offloading decision and scheduling decision, which will be solved with a DRL-based algorithm. The two subproblems are analyzed in the following subsection A and subsection B, respectively.

4.1. Resource Allocation Based on Max-Min Fairness with Fixed Task Assignment. With the fixed offloading decision $\bar{\Omega}_l$, $\bar{\mathbf{\Psi}}_l$, and scheduling decision $\bar{\text{O}}_l$ at time slot l, the communication and computing resource allocation of satellite n can be formulated as two separated max-min fairness problems, which are listed as

$$\begin{aligned} \mathscr{P}1: \min_{C_{n,l}^m} \quad & \max_{\forall m} \tau_{n,m}^{\text{trans}}, \\ \text{s.t.} \quad & \sum_{m\in\bar{\mathbf{\Phi}}_{n,l}} C_{n,l}^m \le \bar{C}_{n,l}, \end{aligned} \tag{11}$$

$$\begin{aligned} \mathscr{P}2: \min_{X_{n,l}^m} \quad & \max_{\forall m} \tau_{n,m}^{\text{pro}}, \\ \text{s.t.} \quad & \sum_{m\in\bar{\mathbf{\Theta}}_{n,l}} X_{n,l}^m \le \bar{X}_{n,l}, \end{aligned} \tag{12}$$

where $\tau_{n,m}^{\text{trans}}$ is the transmission latency for a given task of user m associated with satellite n, and it can be calculated after the task being scheduled from transmission queue and assigned with communication resource. $\tau_{n,m}^{\text{pro}}$ is the processing latency a given task of user m processed at satellite n, and it can be obtained after task being scheduled from processing queue and assigned with computing resource. Here, minimize the maximum latency is adopted as an optimization objective, and it is helpful to guarantee the latency constraints of every task.

Take (11) as an example, introduce auxiliary variable χ, the problem can be rewritten as

$$\begin{aligned} \min_{C_{n,l}^m,\chi} \quad & \chi, \\ \text{s.t.} \quad & \chi \ge \frac{T_m}{C_{n,l}^m}, \forall m, \\ & \sum_{m\in\bar{\mathbf{\Phi}}_{n,l}} C_{n,l}^m \le \bar{C}_{n,l}. \end{aligned} \tag{13}$$

Obviously, the objective function and the second constraint of (13) are both convex. Thus, we only need to prove that the first constraint is convex to prove that this problem is convex.

Let $F = T_m/\chi C_{n,l}^m$, the constraint can be rewritten as $F \le 1, \forall m$. Find the second partial derivative of F, and the Hessian matrix of F can be expressed as

$$H_F = \begin{bmatrix} \dfrac{2T_m}{C_{n,l}^m \chi^3} & \dfrac{T_m}{\left(C_{n,l}^m\right)^2 \chi^2} \\ \dfrac{T_m}{\left(C_{n,l}^m\right)^2 \chi^2} & \dfrac{2T_m}{\left(C_{n,l}^m\right)^3 \chi} \end{bmatrix}. \tag{14}$$

With (14), we can easily get that all the principal minor of H_F are nonnegative, and H_F is a positive semidefinite matrix. Thus, the first constraint of (13) is convex, and problem (13) is a convex problem, which can be solved by the dual ascent method.

Construct Lagrangian functions $\mathscr{L}$ as

$$\mathscr{L}\left(C_{n,l}^m, \mu_m, \nu\right) = \chi + \sum_M \mu_m \left(\frac{T_m}{C_{n,l}^m} - \chi\right) + \nu \left(\sum_{m\in\bar{\mathbf{\Phi}}_{n,l}} C_{n,l}^m - \bar{C}_{n,l}\right), \tag{15}$$

where $\mu_m \ge 0$ and $\nu \ge 0$ are Lagrangian multipliers. Then, the dual function of $\mathscr{L}$ will be

$$\mathscr{D}(\mu_m, \nu) = \mathscr{L}\left(C_{n,l}^{m^*}(\mu_m, \nu), \mu_m, \nu\right), \tag{16}$$

where $C_{n,l}^{m^*} = \arg\min_{C_{n,l}^m} \mathscr{L}(C_{n,l}^m, \mu_m, \nu)$. Since the problem defined in (13) is convex, the maximum value of the $\mathscr{D}$ is equivalent to the minimum value of the problem defined in (13). Thus, we can find the optimal solution of problem defined in (13), which is also the solution of problem defined in (11), via the method proposed in Algorithm 1.

In this paper, μ_m and ν are seen to be converged when the value difference is less than 0.001 for 100 consecutive iterations. By continuously iterating the independent variable and Lagrangian multipliers alternately, we can find the optimal solution $C_{n,l}^m$ for communication resource allocation. Similarly, $X_{n,l}^m$ can be obtained.

4.2. Joint Task Offloading, Scheduling, and Resource Allocation. With Algorithm 1, we can obtain the resource allocation solution for each time slot. However, the joint offloading decision and scheduling decision is still a nonconvex problem with integer programming problem, which cannot be tackled directly with traditional methods based on optimization theory. To address the problem with affordable complexity, we model it as an Markov decision process (MDP) problem and propose a DRL-based method to achieve long-term rewards in terms of task completion rate.

The MDP corresponding to the problem defined in (10) can be expressed as

(1) *State.* The states are defined for every time slot, because the scheduling and resource allocation for tasks are managed slot by slot. The state at time slot l can be defined as $h_l = \{P_U(l), P_S(l), T_l, \tilde{\Xi}_l, \tilde{X}_{U,l}, \tilde{C}_l, \tilde{X}_l, Q_{\text{trans},l}, Q_{\text{pro},l}\}$. $P_U(l)$ and $P_S(l)$ denote the location of users and satellites, respectively. $T_l = \{T_{1,l}, T_{2,l}, \cdots, T_{M,l}\}$ denotes the bits of tasks currently waiting to be scheduled. $\tilde{\Xi}_l = \{\tilde{\Xi}_{1,l}, \tilde{\Xi}_{2,l}, \cdots, \tilde{\Xi}_{M,l}\}$, and $\tilde{\Xi}_{m,l} \in \{1, 2, \cdots, N\}$ denotes the satellite associated with user m at time slot l. $\tilde{X}_{U,l} = \{\tilde{X}_{1,l}, \tilde{X}_{2,l}, \cdots, \tilde{X}_{M,l}\}$ and $\tilde{X}_{m,l} \in \{0, 1\}$, $\tilde{X}_{m,l} = 1$ denotes that the local computing resource is occupied at time slot l. $\tilde{C}_l = \{\tilde{C}_{1,l}, \tilde{C}_{2,l}, \cdots, \tilde{C}_{N,l}\}$ denotes the communication resource of satellites occupied by users at time slot l. Similarly, $\tilde{X}_l = \{\tilde{X}_{1,l}, \tilde{X}_{2,l}, \cdots, \tilde{X}_{N,l}\}$ denotes the computing resource of satellites occupied by users at time slot l. $Q_{\text{trans},l} = \{Q_{\text{trans},l}^1, Q_{\text{trans},l}^2, \cdots, Q_{\text{trans},l}^N\}$ and $Q_{\text{pro},l}$

```
Input:
  Lagrangian multipliers: $\mu_m$, $\nu$
Output:
  Resource allocation: $C_{n,l}^m$
1: Initialize $\mu_m$, $\nu$.
2: Repeat
3:        Find $C_{n,l}^{m^*} \longleftarrow \arg\min_{C_{n,l}^m} \mathscr{L}(C_{n,l}^m, \mu_m, \nu)$.
4:        Update $\chi \longleftarrow \max_{\forall m}(T_m / C_{n,l}^{m^*})$.
5:        Compute $d\mathscr{D}/d\mu_m = (d\mathscr{L}/d\lambda_m)(C_{n,l}^{m^*}, \lambda_m, \nu)$,
           $d\mathscr{D}/d\nu = (d\mathscr{L}/d\nu)(C_{n,l}^{m^*}, \lambda_m, \nu)$.
6: Update $\lambda_m \longleftarrow \lambda_m + \alpha(d\mathscr{D}/d\mu_m)$,
          $\nu \longleftarrow \nu + \alpha(d\mathscr{D}/d\nu)$.
7: Until $\mu_m$, $\nu$ converge.
```

Algorithm 1: Resource allocation based on maximum latency minimization (MLMRA).

$= \{Q_{\text{pro},l}^1, Q_{\text{pro},l}^2, \cdots, Q_{\text{pro},l}^N\}$ denote the total bits of tasks that wait for transmission and processing in the queue of satellites, respectively

(2) *Action*. For each time slot l, the action consists of offloading decision, scheduling decision, and resource allocation of user's current tasks. Since we can obtain the resource allocation with Algorithm 1, we only need to define the action space for offloading decision and scheduling decision with lower dimensions. Thus, the action at time slot l can be defined as $a_l = \{\bar{A}_{1,l}, \bar{A}_{2,l}, \cdots, \bar{A}_{M,l}\}$. $\bar{A}_{m,l} = \{A_{\text{off}}, A_{\text{exe}}\}$, in which $A_{\text{off}} \in \{0, 1, \cdots, Z\}$ denotes the satellite that will handle the current task of user m at time slot l, and $A_{\text{exe}} \in \{0, 1\}$, $A_{\text{exe}} = 1$ denotes that the current task of user m will be scheduled from the queue at time slot l. Otherwise, the task will be kept on waiting in the queue for scheduling. With a specific action a_l, the offloading decision and scheduling decision of all current tasks at time slot l can be obtained correspondingly

(3) *Transition Probability*. For MDP, the transition probability from one state to another is needed for any action a_l. However, it is difficult to get the accurate probability for all of states h_l and actions a_l, because the states space and action space are too large. In this paper, a method based on model-free DRL is considered

(4) *Reward*. To maximize the completion rate of tasks, the reward $R(h_l, a_l)$ at time slot l with state h_l and action a_l is defined as

$$R(h_l, a_l) = \sum_{k \in \bar{O}_l} \left(R_p - \tau_k\right) + dR_d, \tag{17}$$

where τ_k denotes the latency defined in (8) or (9) for task k at time slot l. R_p is a constant value that makes the $R_p - \tau_k$ positive. R_d is a positive completion reward, and d denotes the number of tasks which are completed within the latency constraints in time slot l.

Given the action policy π, value function $V(h \mid \pi)$, which can be used to evaluate the long-term performance of the policy π, is defined as

$$V(h \mid \pi) = \mathbb{E}\left[\sum_l \gamma^l R(h_l, a_l) \mid h_0 = h, \pi\right], \tag{18}$$

where γ denotes the discount factor, and the value function can be seen as an expectation of completion rate defined in (10) with $\gamma = 1$. Thus, the optimal policy π^* can be expressed as

$$\pi^*(h) = \arg\max_a \left[R(h, a) + \sum_{\bar{h}} \gamma V\left(\bar{h} \mid \pi^*\right)\right]. \tag{19}$$

where state $\bar{h}$ can be obtained with action a and state h. In this paper, deep Q-network (DQN) [28, 29], which is composed of target network and main network, is adopted to obtain the target Q-value $Q^*(h, a)$. Moreover, the approximated Q-function $Q(h, a; \theta)$ will approach $Q^*(h, a)$ via training process by minimizing the loss function, which can be defined as $L(\theta) = E[(Q^*(h, a) - Q(h, a; \theta))^2]$. And θ is the weight of network. The detailed description and analysis for the processes of DQN can be found in [23]. The proposed joint task offloading, scheduling, and resource allocation (JTOSRA) approach for collaborative computing among LEO satellites is shown in Algorithm 2, where G denotes maximum of training step, and ζ denotes the experience replay buffer. ε-greedy policy is utilized to balance the exploration and utilization of models [29], and ε will decay from 1.0 to 0.001 through 20000 steps.

Generally, it is hard to obtain the accurate computational complexity of a DRL-based algorithm. In Algorithm 2, the computational complexity of the DQN network mainly depends on the number of users and the network structure

```
Input:
  IoT terminal information: $P_U(l)$, $\tilde{X}_{U,l}$, $T_l$
  Satellite information: $P_S(l)$, $\tilde{\Xi}_l$, $\tilde{C}_l$, $\tilde{X}_l$
  Queue information: $Q_{trans,l}$, $Q_{pro,l}$
Output:
  Offloading and scheduling decisions: $\bar{\Omega}_l$, $\bar{\mathbf{\Psi}}_l$, $\bar{O}_l$
  Resource allocation: $C_l$, $X_l$
1: Initialize network with $\gamma$, $\varepsilon$ and $\zeta$.
2: Initialize state $h_l$
3: While $l < G$
4:     Select an action $a_l$ according to $\varepsilon$-greedy policy.
5:     Allocate resource according to Algorithm 1.
6:     Calculate reward $R(h_l, a_l)$.
7:     Update next state $h_{l+1}$.
8:     Save $(h_l, a_l, R(h_l, a_l), h_{l+1})$, and update $\zeta$.
9:     Update $\theta$.
10:      $l$ ++.
11: End while
```

Algorithm 2: Joint task offloading, scheduling, and resource allocation based on DQN.

of neural network utilized by DQN. All layers of the neural network used in this algorithm are fully connected layers, and the number of input parameters of the network is determined by the state space, which can be represented as n_0. Assuming that the number of neural network layers is J, and the number of neurons in the j-th layer is expressed by n_j, the computational complexity for the DQN network can be expressed as $O(G \times (\sum_{j=0}^{J} n_j n_{j+1}))$ [30].

5. Simulation Results

5.1. Simulation Configurations. Simulation parameters are listed in Table 1. In the simulation, we focus on the users located in a specific area covered by the LEO satellites. The simulation time starts at 0 : 00 on October 1, 2020, and the Greenwich hour angle θ_{g_0} at this moment is 10.2. The communication capacity of satellite is set to 10 Gbps. The CPU cycle needed for processing is set to 1000 cycle/-bit [31]. And we set the satellite computing capacity and local computing capacity to 10 GC/s [32] and 1.5 GC/s [33], respectively.

5.2. Convergence of JTOSRA. Figure 2 shows the convergence of the loss function. It can be seen that the loss function defined in $L(\theta)$ will converge when the training steps increase. As shown in Figure 3, the completion rate of tasks will also increase during the training processes. Figures 2 and 3 demonstrate that the JTOSRA based on DQN is applicable for the problem formulated in Section 3. Though a large amount of training steps is needed to achieve convergence, the training processes will only be implemented in the initial phase of the LEO network. Once the convergence is achieved, joint task offloading and scheduling decisions can be made step by step with low complexity. Moreover, the training processes can be implemented offline via pretraining processes to decrease the complexity further.

Table 1: Simulation parameters.

Parameter	Symbol	Value
Scene parameters		
Task size	T_m	8×10^4-1.2×10^5 bit
Rate of task arrival	λ	0.05
Latency constraint of task	$\tau_{\max}$	200 ms
Length of time slot	ρ	10 ms
Number of satellite orbits	P	18
Number of satellites per orbit	S	40
Height of LEO satellite	H	1200 km
Inclination of satellite orbit	i	87.9°
Phase factor of orbit	F	1
Minimum elevation angle of user	$\sigma_{\min}$	20°
Greenwich hour angel	θ_{g_0}	10.2°
CPU cycle needed for processing	ξ	1000 cycles/bit
Satellite communication capacity	C	10 Gbps
Satellite computing capacity	X_S	10 GC/s
Local computing capacity	X_U	1.5 GC/s
Algorithm parameters		
Step size of dual ascent	α	0.001
Maximum training episode	G	150000
Size of replay buffer	ζ	20000
Observation size	O_b	5000
Discount factor	γ	0.95
Positive reward	R_p	5
Completion reward	R_d	15
Learning rate	ι	0.001

5.3. Performance Analysis. To analyze the performance of MLMRA with fixed task assignment, two reference schemes are adopted and compared in terms of completion rate. The referred resource allocation scheme is listed as

(i) *Average Resource Allocation (ARA)*. The resource will be allocated evenly to tasks

(ii) *Resource Allocation Based on Latency Minimization (LMRA)*. The resource will be allocated to tasks by minimizing the sum latency of tasks

Figure 4 shows the influence of the number of tasks on the completion rate with different resource allocation method. It can be seen that the completion rate of the task will decrease along with the increase of the number of tasks. Moreover, the MLMRA performs better than the LMRA, and the ARA algorithm performs the worst. This shows that the MLMRA can allocate resources more equitably and minimize the maximum latency of the task. This is because that the LMRA focuses on reducing the sum latency of tasks, while the MLMRA will allocate more resources to tasks that are difficult to be completed within the limited latency. In general, the proposed MLMRA algorithm can effectively improve the completion rate of the task with fairness among tasks.

To evaluate the performance of the JTOSRA, the following two algorithms are introduced for task assignment:

(i) *Random*. Tasks will be offloaded and scheduled randomly, and resources will be allocated according to LMRA and MLMRA. And in Figures 5–7, the methods are labeled as random-LMRA and random-MLMRA, respectively

(ii) *Simulated Annealing (SA)*. Tasks will be offloaded and scheduled through the SA algorithm, and resources will be allocated by LMRA and MLMRA. In Figures 5–7, the methods are labeled as SA-LMRA and SA-MLMRA, respectively

In addition, in order to ensure the validity of simulation results, each point in the figures is obtained by taking the average value over multiple tests, and each test lasts for 200000 slots. Figure 5 shows the completion rate performance of the algorithms as the number of users increases. With the increase of the number of users, the number of tasks waiting for scheduling will increase, and the shortage of resources will lead to the decline of the task completion rate. Obviously, the proposed JTOSRA algorithm performs better than the SA algorithm. And for the JTOSRA, the task completion rate decreases slower than that of the SA algorithm as the number of users increases. This is because the SA algorithm tends to fall into local optimal solutions, resulting in poor algorithm performance. On the other hand, traditional algorithms such as the SA algorithm can only optimize the decision for a specific time slot, but cannot continuously optimize the offloading and scheduling decisions for multiple time slots, and the algorithm needs to iterate at each step, which brings high time cost. However, the proposed JTOSRA algorithm can continue to accumulate experience in the decision-making process to optimize the

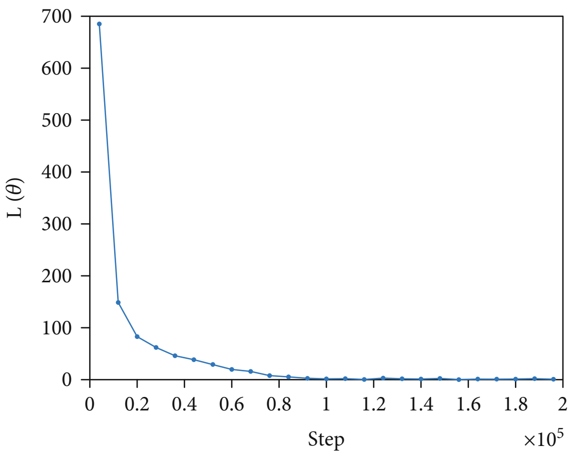

FIGURE 2: Convergence process of loss.

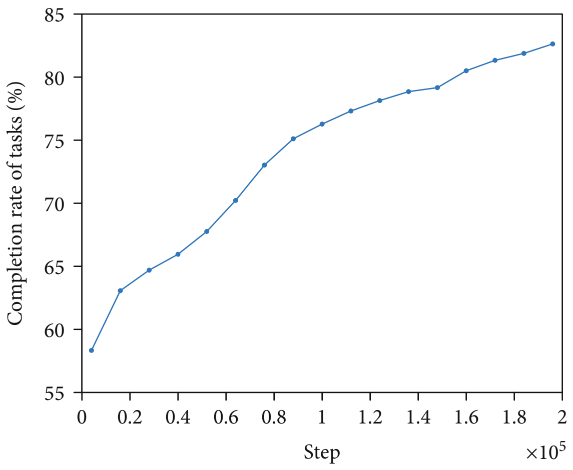

FIGURE 3: Convergence process of completion rate.

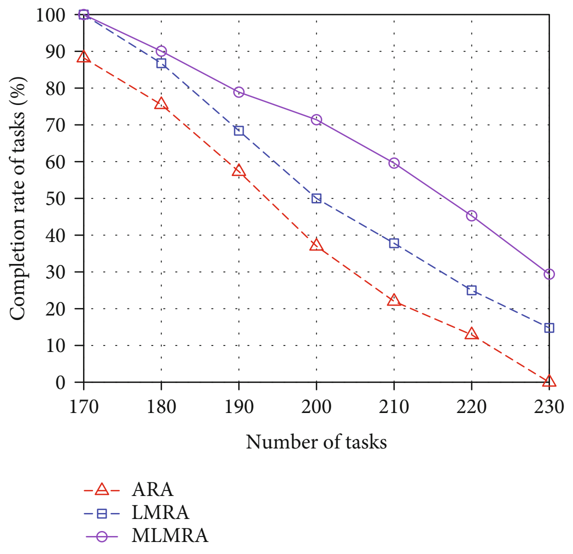

FIGURE 4: Completion rate of tasks vs. number of tasks.

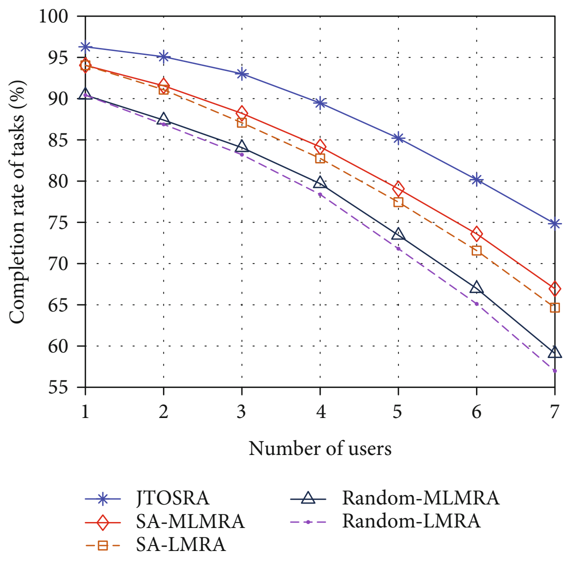

FIGURE 5: Completion rate of tasks vs. number of users.

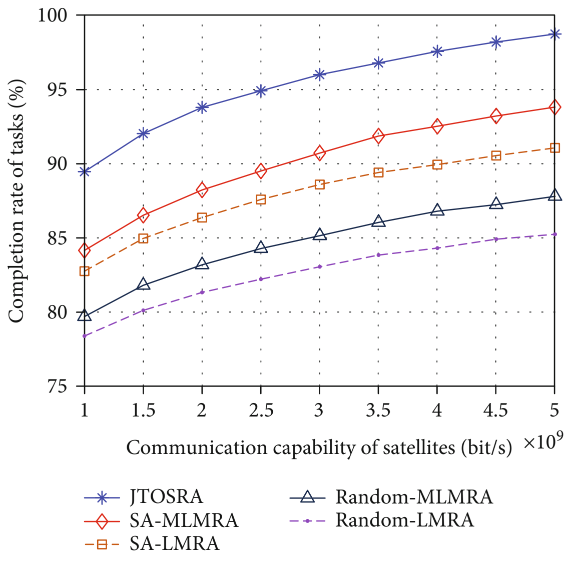

FIGURE 6: Completion rate of tasks vs. communication capability of satellites.

completion rate of tasks. In addition, MLMRA performs better than LMRA.

In Figure 6, satellite communication capability is adopted as variable to investigate the performance of the proposed algorithm. It can be seen that the increase of satellite communication resources, which means that more communication resource can be allocated to tasks, will lead to the increasing of completion rate. But the increasing rate of the curve decreases with the rise of satellite communication resources. This is because that communication resource is the main factor influencing the latency of tasks when the amount of communication resources is small. When the amount of communication resources of satellites increases

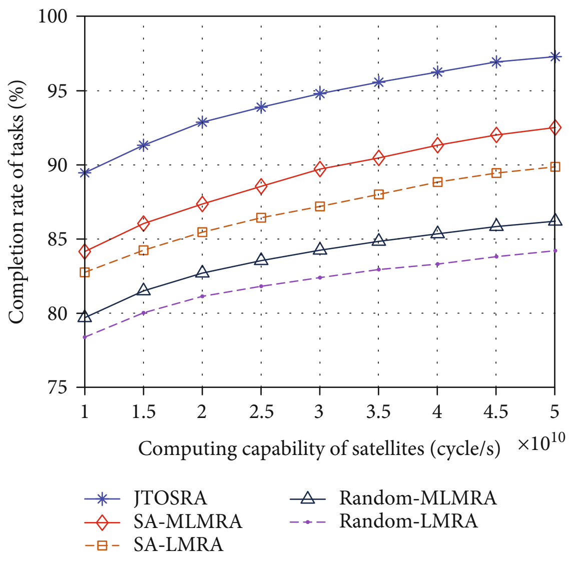

FIGURE 7: Completion rate of tasks vs. computing capability of satellites.

to a certain value, the computing resources of satellites will become the dominant factor, which will mainly affect the latency of tasks. In addition, the performance of the JTOSRA algorithm is still better than that of the other two algorithms, and the performance of the MLMRA algorithm is better than that of the LMRA algorithm. Similarly, the completion rate with respect to satellite computing capability is shown in Figure 7. The variation trend of each curve in the figure is close to that in Figure 6.

6. Conclusion

In this paper, collaborative computing and resource allocation for LEO satellite networks are investigated. A framework for collaborative computing among LEO satellites with varying topology is proposed, and the joint task offloading, scheduling, and multidimensional resource allocation problem is divided into two subproblems with low complexity. JTOSRA based on DRL and max-min fairness is proposed to solve the problems, and simulation results demonstrate that the JTOSRA outperforms the referred schemes in terms of task completion rate.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61971054 and Grant 62171052 and Science and Technology on Information Transmission and Dissemination in

Communication Networks Laboratory Foundation under Grant HHX21641X002.

References

[1] O. Kodheli, E. Lagunas, N. Maturo et al., “Satellite communications in the new space era: a survey and future challenges,” *IEEE Communications Surveys Tutorials*, vol. 23, no. 1, pp. 70–109, 2021.

[2] R. Xie, Q. Tang, Q. Wang, X. Liu, F. R. Yu, and T. Huang, “Satellite-terrestrial integrated edge computing networks: architecture, challenges, and open issues,” *IEEE Network*, vol. 34, no. 3, pp. 224–231, 2020.

[3] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, “Mobile edge computing: a survey,” *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450–465, 2018.

[4] C. Adams, A. Spain, J. Parker, M. Hevert, J. Roach, and D. Cotten, “Towards an integrated GPU accelerated SoC as a flight computer for small satellites,” in *2019 IEEE Aerospace Conference*, pp. 1–7, Big Sky, MT, USA, 2019.

[5] L. Yan, S. Cao, Y. Gong et al., “Satec: a 5G satellite edge computing framework based on microservice architecture,” *Sensors*, vol. 19, no. 4, p. 831, 2019.

[6] Z. Zhang, W. Zhang, and F. 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.

[7] I. F. Akyildiz and A. Kak, “The internet of space things/cubesats,” *IEEE Network*, vol. 33, no. 5, pp. 212–218, 2019.

[8] S. Cioni, R. De Gaudenzi, O. Del Rio Herrero, and N. Girault, “On the satellite role in the era of 5G massive machine type communications,” *IEEE Network*, vol. 32, no. 5, pp. 54–61, 2018.

[9] J. Liu, X. Du, J. Cui, M. Pan, and D. Wei, “Task-oriented intelligent networking architecture for the space?air?ground?aqua integrated network,” *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5345–5358, 2020.

[10] C. Jiang and X. Zhu, “Reinforcement learning based capacity management in multi-layer satellite networks,” *IEEE Transactions on Wireless Communications*, vol. 19, no. 7, pp. 4685–4699, 2020.

[11] Y. Su, Y. Liu, Y. Zhou, J. Yuan, H. Cao, and J. Shi, “Broadband leo satellite communications: architectures and key technologies,” *IEEE Wireless Communications*, vol. 26, no. 2, pp. 55–61, 2019.

[12] I. Del Portillo, B. G. Cameron, and E. F. Crawley, “A technical comparison of three low earth orbit satellite constellation systems to provide global broadband,” *Acta Astronautica*, vol. 159, pp. 123–135, 2019.

[13] N. U. L. Hassan, C. Huang, C. Yuen, A. Ahmad, and Y. Zhang, “Dense small satellite networks for modern terrestrial communication systems: benefits, infrastructure, and technologies,” *IEEE Wireless Communications*, vol. 27, no. 5, pp. 96–103, 2020.

[14] M. Sheng, D. Zhou, R. Liu, Y. Wang, and J. Li, “Resource mobility in space information networks: opportunities, challenges, and approaches,” *IEEE Network*, vol. 33, no. 1, pp. 128–135, 2019.

[15] A. Papa, T. de Cola, P. Vizarreta, M. He, C. Mas-Machuca, and W. Kellerer, “Design and evaluation of reconfigurable sdn leo constellations,” *IEEE Transactions on Network and Service Management*, vol. 17, no. 3, pp. 1432–1445, 2020.

[16] N. Cheng, F. Lyu, W. Quan et al., “Space/aerial-assisted computing offloading for IoT applications: a learning-based approach,” *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 5, pp. 1117–1129, 2019.

[17] B. Cao, J. Zhang, X. Liu et al., “Edge-cloud resource scheduling in space-air-ground integrated networks for internet of vehicles,” *IEEE Internet of Things Journal*, p. 1, 2021.

[18] 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, 2019.

[19] F. Wang, D. Jiang, S. Qi, C. Qiao, and H. Song, “Fine-grained resource management for edge computing satellite networks,” in *2019 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, Waikoloa, HI, USA, 2019.

[20] M. LiWang, S. Dai, Z. Gao, X. Du, M. Guizani, and H. Dai, “A computation offloading incentive mechanism with delay and cost constraints under 5G satellite-ground IoV architecture,” *IEEE Wireless Communications*, vol. 26, no. 4, pp. 124–132, 2019.

[21] J. Jiao, Y. Sun, S. Wu, Y. Wang, and Q. Zhang, “Network utility maximization resource allocation for noma in satellite-based internet of things,” *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3230–3242, 2020.

[22] B. Denby and B. Lucia, “Orbital edge computing: machine inference in space,” *IEEE Computer Architecture Letters*, vol. 18, no. 1, pp. 59–62, 2019.

[23] G. Cui, Y. Long, L. Xu, and W. Wang, “Joint offloading and resource allocation for satellite assisted vehicle-to-vehicle communication,” *IEEE Systems Journal*, vol. 15, no. 3, pp. 3958–3969, 2021.

[24] S. Zhang, G. Cui, Y. Long, and W. Wang, “Optimal resource allocation for satellite-aided collaborative computing among multiple user pairs,” *International Journal of Satellite Communications and Networking*, vol. 39, no. 5, pp. 500–508, 2021.

[25] Z. Song, Y. Hao, Y. Liu, and X. Sun, “Energy efficient multi-access edge computing for terrestrial-satellite internet of things,” *IEEE Internet of Things Journal*, vol. 8, no. 18, pp. 14202–14218, 2021.

[26] H. Liao, Z. Zhou, X. Zhao, and Y. Wang, “Learning-based queue-aware task offloading and resource allocation for space–air–ground-integrated power iot,” *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5250–5263, 2021.

[27] Q. Tang, Z. Fei, B. Li, and Z. Han, “Computation offloading in leo satellite networks with hybrid cloud and edge computing,” *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 9164–9176, 2021.

[28] I. Osband, C. Blundell, A. Pritzel, and B. Van Roy, “Deep exploration via bootstrapped dqn,” *Advances in neural information processing systems*, pp. 4026–4034, 2016.

[29] V. Mnih, K. Kavukcuoglu, D. Silver et al., “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[30] X. Huang, S. Leng, S. Maharjan, and Y. Zhang, “Multi-agent deep reinforcement learning for computation offloading and interference coordination in small cell networks,” *IEEE Transactions on Vehicular Technology*, p. 1, 2021.

[31] 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, 2018.

[32] S. Yang, S. Cao, J. Wei, Y. Zhao, and L. Yan, "Space-based computing platform based on soc fpga," in *2019 IEEE World Congress on Services (SERVICES)*, Milan, Italy, 2019.

[33] J. Zhang, H. Guo, J. Liu, and Y. Zhang, "Task offloading in vehicular edge computing networks: a load-balancing solution," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 2092–2104, 2020.