symmetry

Article

Joint Offloading Decision and Resource Allocation in Mobile Edge Computing-Enabled Satellite-Terrestrial Network

Minglei Tong [1,2], Xiaoxiang Wang [1,2,*], Song Li [3] and Liang Peng [1,2]

[1] School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China; tongminglei@bupt.edu.cn (M.T.); 2020010070@bupt.edu.cn (L.P.)
[2] Key Laboratory of Universal Wireless Communications, Ministry of Education, Beijing University of Posts and Telecommunications, Beijing 100876, China
[3] School of Information and Control Engineering, China University of Mining and Technology, Xuzhou 221116, China; lisong@cumt.edu.cn
* Correspondence: cpwang@bupt.edu.cn

Abstract: With the development of satellite-terrestrial network (STN), mobile edge computing (MEC) servers are deployed at low orbit earth (LEO) satellites to provide computing services for user devices (UEs) in areas without terrestrial network coverage. There is symmetry between satellite networks and terrestrial networks, but there is asymmetry between their resources. Computing resources of satellites' MEC servers may not be enough. The satellite-terrestrial cooperation is promising, where a satellite migrates tasks to a base station (BS) in an adjacent area, thus utilizing computing resources of the BS's MEC server. Although there are some studies on computation offloading in STN, few studies consider a satellite as both a relay and a computing unit to assist UEs in computing tasks. This paper proposes a joint offloading decision and resource allocation scheme in MEC-enabled STN, which minimizes the completion delay of all UEs' indivisible tasks. Firstly, the optimization problem is formulated and decomposed. Then, the proposed scheme based on potential game and the Lagrange multiplier method makes UEs' task offloading decisions and allocates the satellite's and the BS's computing resources, thus obtaining the optimal solution through continuous iterations. Finally, the simulation results validate that the proposed scheme can obtain better gain than other baseline schemes.

Keywords: satellite-terrestrial network; mobile edge computing; satellite-terrestrial cooperation; offloading decision; resource allocation

check for updates

Citation: Tong, M.; Wang, X.; Li, S.; Peng, L. Joint Offloading Decision and Resource Allocation in Mobile Edge Computing-Enabled Satellite-Terrestrial Network. *Symmetry* **2022**, *14*, 564. https://doi.org/10.3390/sym14030564

Academic Editor: Boris Malomed

Received: 11 February 2022
Accepted: 9 March 2022
Published: 12 March 2022

1. Introduction

With the development of networks and the popularization of user devices (UEs), people can easily access networks and obtain desired services. However, the number of service requests generated per moment increases significantly. Moreover, more and more delay-sensitive and computation-intensive applications, such as autonomous driving, virtual reality, augmented reality, cloud robotics, and remote surgery, etc. [1] have been integrated into people's daily life. UE's computing capabilities are limited, so they cannot complete the tasks of these applications on their own within low delay [2]. High task completion delay will affect the quality of user experience seriously.

Cloud servers have strong computing capabilities. However, they are deployed at the core network layer and are too far away from UEs. Therefore, transmitting tasks to cloud servers for computing will bring extremely high delay and extra transmission energy consumption [3]. Mobile edge computing (MEC) proposed by the European Telecommunication Standardization Organization (ETSI) has attracted people's attention. It makes the information technology (IT) service environment and cloud computing capacity sink from the cloud to the wireless access network, enforces the computing capacity to the edge of mobile networks, and reduces the response delay effectively [4]. Users can offload tasks to nearby network edges for computing via terrestrial networks, e.g., cellular networks [5].

To implement MEC in cellular networks, an MEC server with computing capability is deployed at a base station (BS). The BS can receive tasks offloaded by UEs within its coverage area and use its MEC server to compute tasks [6]. This not only makes up for UEs' computing capabilities, prolongs their battery life, but also achieves better service [7].

Terrestrial networks cover only about 20% of the earth's land area, and less than 6% of its surface [8]. Satellite networks can complement and expand terrestrial networks, so as to achieve seamless global coverage, e.g., low earth orbit (LEO) satellite networks can provide users with all-weather stable and reliable services in both densely populated areas and rural areas [9]. In order to expand the service scope and provide reliable and on-demand services, etc., the convergence of satellite networks and terrestrial networks, i.e., satellite-terrestrial networks (STNs) has become a research hotspot [10]. STN has advantages, e.g., wide coverage, strong robustness, etc. This paper introduces MEC into STN, i.e., the study is carried out in MEC-enabled STN.

For the areas that are not covered by terrestrial networks, users can rely on satellite networks to obtain services. Comparing with terrestrial networks, LEO satellite networks can provide high-capacity backhaul, large coverage, and more flexible network access services [11]. In addition, LEO satellite networks have advantages, e.g., low transmission loss and low transmission delay, which can ensure UEs' quality of service (QoS) [12]. MEC servers can be deployed at LEO satellites, so users can obtain MEC services by communicating with LEO satellites [13]. UEs can offload tasks to LEO satellites for computing. Comparing with sending tasks to the remote cloud, the delay is improved significantly [14].

It should be noted that there is symmetry between satellite networks and terrestrial networks, but there is asymmetry between their resources. Comparing with MEC servers of BSs, MEC servers of LEO satellites have fewer computing resources. When overmany UEs request tasks simultaneously or overmany computing resources are required, LEO satellites' MEC servers may not provide sufficient computing resources for all UEs [15]. The ideal gain cannot be obtained only by offloading tasks to LEO satellites for computing. This situation highlights the necessity of satellite-terrestrial cooperation. LEO satellites migrate some tasks to BSs in adjacent areas, utilizing computing resources of the BSs' MEC servers to compute tasks.

In addition, computation offloading will cause additional delay [16], such as the delay of transmitting tasks on the uplink, which does not exist in local computing. Therefore, reasonable task offloading decision and computing resource allocation are very important in satellite-terrestrial cooperation, i.e., the cooperation between the LEO satellite in the local area and the terrestrial BS in the adjacent area, which assists UEs in computing tasks.

In this paper, the STN consists of UEs, an LEO satellite, and a BS. Considering tasks of UEs in a rural area. The LEO satellite and the BS work as computing nodes, which can assist UEs computing tasks through satellite-terrestrial cooperation. This paper proposes a joint offloading decision and resource allocation scheme in MEC-enabled STN, which minimizes the completion delay of all UEs' indivisible tasks.

The main contributions of this paper are as follows.

- The system model of MEC-enabled STN is designed, which utilizes the satellite-terrestrial cooperation to provide MEC services for UEs in the rural area without coverage of terrestrial networks. The joint offloading decision and resource allocation problem to minimize the task completion delay of all UEs is formulated, and then decomposed into the task offloading decision problem and computing resource allocation problem.
- A joint offloading decision and resource allocation scheme is proposed. The computing resource allocation problem is first decomposed into two sub-problems, which is according to the LEO satellite and the BS, respectively. Both sub-problems are convex and can be solved by Lagrange multiplier method. The task offloading decision problem is considered as a multi-UEs task offloading game, which is proved to be an

exact potential game (EPG) and reach a Nash equilibrium (NE) within a finite number of iterations.

- The performance of the proposed scheme is evaluated by extensive simulations. The simulation results denote that the proposed scheme can achieve better performance gain compared with other baseline schemes.

The rest of this paper is structured as follows. The related work is presented in Section 2. Section 3 describes the system model. The UEs' task completion delay minimization problem is formulated in Section 4. The solving process and the proposed scheme are exhibited in Section 5. In Section 6, the simulation results and discussion are presented. The conclusion is provided in Section 7.

2. Related Work

At present, there are some studies on computation offloading in STN. Wang et al. [17] proposed a double computation offloading algorithm based on STN, in which tasks are assigned to edge servers at the lowest cost, so as to minimize system energy consumption and reduce task offloading delay. Zhang et al. [18] proposed the satellite MEC, which allows UEs without a nearby MEC server to enjoy MEC services through satellite links. It also designed a cooperative computation offloading model to realize STN parallel computing, which minimizes user perceived delay and system energy consumption by optimizing task scheduling. Cheng et al. [19] designed a space–air–ground integrated network (SAGIN) architecture, which includes Internet of Things (IoT) devices, unmanned aerial vehicles (UAVs), and satellites. UAVs have edge servers, and satellites provide access to cloud servers. A learning-based computing offloading approach was proposed, thus converging fast and reducing total cost. Xie et al. [20] proposed an architecture named satellite-terrestrial integrated edge computing network (STECN), which includes edge computing clusters, terrestrial MEC platforms, and satellite MEC platforms. The former two are closer to UEs, but have the limited service scope. The latter has the large service scope, but is further from UEs, the transmission delay is higher. Zhang et al. [21] designed a double intelligent satellite-terrestrial integrated network. The satellite MEC server assists the terrestrial MEC server to compute tasks offloaded by users. A task migration strategy based on greedy algorithm was proposed, so as to achieve load balancing and reduce system delay. Chen et al. [22] proposed a Ka/Q-band satellite-terrestrial integrated network (STIN), where satellites are equipped with transparent repeaters without computing capabilities. Satellites receive tasks offloaded by Internet of remote things (IoRT) devices, and then forward tasks to ground stations. A learning-based computation offloading approach was proposed, thus improving the total reward, increasing the number of tasks satisfied, and decreasing the power consumption. Song et al. [23] proposed an MEC framework for the satellite-terrestrial IoT. With the assistance of satellite-terrestrial terminals, mobile devices of IoT can offload tasks to LEO satellites for computing. It also proposed an energy-efficient algorithm of computation offloading and resource allocation, so as to minimize the weighted sum of energy consumption of IoT mobile devices. Pang et al. [24] proposed a coded computation offloading strategy (CCOS) in STIN, thus migrating ultra-dense computing tasks on the ground to distributed satellite constellations. The access satellite receives tasks and transmits them to other satellites without providing computing services. The effect of stragglers is alleviated, and the delay-energy tradeoff cost is minimized. Wang et al. [25] designed a satellite-terrestrial double edge computing network architecture, where the computing capability of overloaded terrestrial MEC server is supplemented by LEO satellite MEC server. The former executes tasks in serial, the latter executes tasks in parallel. The local computing of IoT nodes is not considered. A joint optimization method of BS task execution order and satellite resource allocation was proposed, which increases the profit of MEC resource providers, and meanwhile satisfies QoE requirements of tasks. However, as far as we know, there are few studies on satellite-terrestrial cooperation, where a satellite as both a relay and a computing unit, assisting UEs in computing tasks.

3. System Model

In this section, the network model, the channel model, and the computation model are introduced in detail and sequence.

3.1. Network Model

As shown in Figure 1, this paper considers a rural area that is not covered by terrestrial networks. There is an LEO satellite with an MEC server in space that can cover this area. The computing capability of the MEC server deployed at the LEO satellite is weaker than that of the MEC server deployed at the BS. N UEs are randomly distributed in this area, and the set is expressed as $\mathcal{N}$. These UEs are very small aperture terminals (VAST), which are equipped with directional antennas, so UEs can communicate with the LEO satellite directly. Each UE requests an indivisible task with two attributes, e.g., UE i requests task i (c_i, a_i), c_i is the computing intensity, i.e., the computing resources required to compute 1 bit of task i (cycles per bit), and a_i is the data size of task i (bits). UEs can offload tasks to the LEO satellite, utilizing its MEC server to compute tasks. There is a BS deployed with an MEC server in an adjacent area, and the BS connects to a near gateway by wired connection. In addition to computing tasks by itself, the LEO satellite can migrate some tasks to the BS and make use of its MEC server to compute tasks. Different UEs have different computing capabilities. MEC servers have more computing resources than UEs, but their computing resources are limited. UE i has three strategies to complete its own task. UE i computes task i locally, i.e., local computing. UE i offloads task i to the LEO satellite, task i is computed by the LEO satellite's MEC server, i.e., LEO satellite edge computing. UE i offloads task i to the LEO satellite, and the LEO satellite migrates task i to the BS, task i is computed by the BS's MEC server, i.e., LEO satellite-assisted terrestrial edge computing.

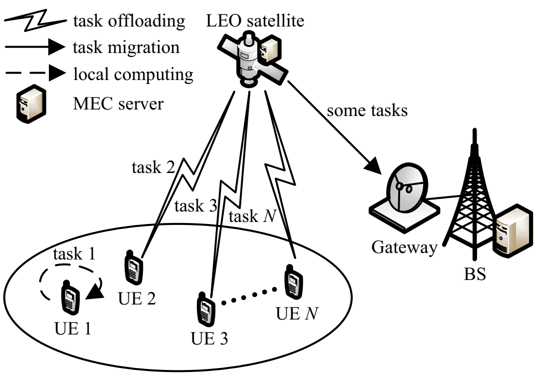

Figure 1. Network model.

3.2. Channel Model

To avoid interference among UEs, the orthogonal frequency division multiple access (OFDMA) is used, which allocates a sub-channel with the same bandwidth to each UE. The satellite-terrestrial transmission link between UEs and the satellite uses Ka band [26]. Since line-of-sight (LOS) transmission is mainly considered, the link uses the Ricean channel and takes into account free-space path loss (FSPL) and additive white Gaussian noise (AWGN). The achievable data rate of the link between UE i and the LEO satellite is

$$r_{i,\mathrm{S}} = B_{i,\mathrm{S}}\log_2\left(1 + \frac{p_i g_i G_\mathrm{r} H_{i,\mathrm{S}}}{N_{0_{i,\mathrm{S}}}}\right), \quad (1)$$

$B_{i,\mathrm{S}}$ is the bandwidth of the sub-channel (Hertz), p_i is the transmitting power of UE i (Watt), g_i is the transmitting antenna gain of UE i, G_r is the receiving antenna gain of the LEO satellite, $H_{i,\mathrm{S}}$ is the channel gain of the link, and $N_{0_{i,\mathrm{S}}}$ is the noise power of the link. $G_\mathrm{r} \approx (4D_\mathrm{RA} f_\mathrm{u}/c)^2$, D_RA is the diameter of the LEO satellite receiving antenna (meter) [27], and f_u is the carrier frequency of the uplink (Hertz). $H_{i,\mathrm{S}} = (h_{i,\mathrm{S}})^2$, channel coefficient

$h_{i,\mathrm{S}} = \rho_{i,\mathrm{S}} / \sqrt{L^{\mathrm{f}}_{i,\mathrm{S}}}$, $\rho_{i,\mathrm{S}}$ is the Ricean stochastic variable, $L^{\mathrm{f}}_{i,\mathrm{S}}$ is FSPL, $L^{\mathrm{f}}_{i,\mathrm{S}} = (4\pi d_{i,\mathrm{S}} f_{\mathrm{u}}/c)^2$, c is the speed of light (meter per second), $d_{i,\mathrm{S}}$ is the distance between UE i and the LEO satellite (meter). As shown in Figure 2, $d_{i,\mathrm{S}} = \sqrt{(R_{\mathrm{e}})^2 + (R_{\mathrm{e}} + h_{\mathrm{s}})^2 - 2R_{\mathrm{e}}(R_{\mathrm{e}} + h_{\mathrm{s}})\cos\theta_{\mathrm{ec}}}$, where R_{e} is the earth radius, h_{s} is the orbital height of the LEO satellite, and θ_{ec} is the geocentric angle corresponding to the area covered by the LEO satellite. $\theta_{\mathrm{ec}} = \arccos[R_{\mathrm{e}} \cos\theta_{\mathrm{u}} / (R_{\mathrm{e}} + h_{\mathrm{s}})] - \theta_{\mathrm{u}}$, θ_{u} is the elevation angle between UE i on the ground and the LEO satellite in space [28].

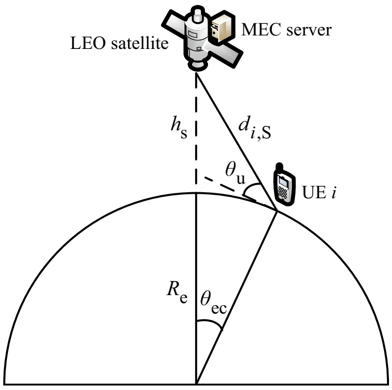

Figure 2. Illustration of $d_{i,\mathrm{S}}$.

The satellite-terrestrial transmission link between the LEO satellite and the gateway uses Ka band [26]. Since LOS transmission is mainly considered, the link uses the Ricean channel and considers FSPL and AWGN. The achievable data rate of the link between the LEO satellite and the gateway is

$$r_{\mathrm{S,G}} = B_{\mathrm{S,G}} \log_2\left(1 + \frac{P G_{\mathrm{t}} G_{\mathrm{g}} H_{\mathrm{S,G}}}{N_{0_{\mathrm{S,G}}}}\right), \tag{2}$$

$B_{\mathrm{S,G}}$ is the bandwidth of the sub-channel (Hertz), P is the transmitting power of the LEO satellite (Watt), G_{t} is the transmitting antenna gain of the LEO satellite, G_{g} is the receiving antenna gain of the gateway, $H_{\mathrm{S,G}}$ is the channel gain of the link, $N_{0_{\mathrm{S,G}}}$ is the noise power of the link. $G_{\mathrm{t}} \approx (4D_{\mathrm{TA}} f_{\mathrm{d}}/c\,)^2$, D_{TA} is the diameter of the LEO satellite transmitting antenna (meter) [27], and f_{d} is the carrier frequency of the downlink (Hertz). Similarly, $G_{\mathrm{g}} \approx (4D_{\mathrm{g}} f_{\mathrm{d}}/c\,)^2$, D_{g} is the diameter of the gateway satellite antenna (meter). $H_{\mathrm{S,G}} = (h_{\mathrm{S,G}})^2$, channel coefficient $h_{\mathrm{S,G}} = \rho_{\mathrm{S,G}} / \sqrt{L^{\mathrm{f}}_{\mathrm{S,G}}}$, $\rho_{\mathrm{S,G}}$ is the Ricean stochastic variable, $L^{\mathrm{f}}_{\mathrm{S,G}}$ is FSPL, $L^{\mathrm{f}}_{\mathrm{S,G}} = (4\pi d_{\mathrm{S,G}} f_{\mathrm{d}}/c)^2$, $d_{\mathrm{S,G}}$ is the distance between the LEO satellite and the gateway (meter). $d_{\mathrm{S,G}}$ is computed in the same way as $d_{i,\mathrm{S}}$, except that θ_{ec} is computed with θ_{g} instead of θ_{u}, θ_{g} is the elevation angle between the gateway on the ground and the LEO satellite in space.

3.3. Computation Model

The indicator variables $x_{i,0}, x_{i,\mathrm{S}}, x_{i,\mathrm{B}} \in \{0,1\}, \forall i \in \mathcal{N}$ are set to represent task offloading decisions. When UE i chooses local computing, $x_{i,0} = 1$, otherwise, $x_{i,0} = 0$. When UE i chooses LEO satellite edge computing, $x_{i,\mathrm{S}} = 1$, otherwise, $x_{i,\mathrm{S}} = 0$. When UE i chooses LEO satellite-assisted terrestrial edge computing, $x_{i,\mathrm{B}} = 1$, otherwise, $x_{i,\mathrm{B}} = 0$.

When $x_{i,0} = 1$, the delay is the delay that UE i computes task i.

$$D_{i,0} = \frac{c_i a_i}{f_{i,0}}, \tag{3}$$

$f_{i,0}$ is the computing capability of UE i (cycles per second).

The size of computing result data is usually much smaller than that of input data. Therefore, the delay of sending result data from the MEC server at the LEO satellite or the BS back to UEs is ignored. However, the propagation delay from the LEO satellite to UEs and from the gateway to the LEO satellite cannot be ignored [29].

When $x_{i,\mathrm{S}} = 1$, the delay includes the delay that UE i offloads task i to the LEO satellite, the propagation delay between UE i and the LEO satellite, and the delay that the LEO satellite's MEC server computes task i.

$$D_{i,\mathrm{S}} = \frac{a_i}{r_{i,\mathrm{S}}} + \frac{2d_{i,\mathrm{S}}}{c} + \frac{c_i a_i}{f_{i,\mathrm{S}}}, \tag{4}$$

$f_{i,\mathrm{S}}$ are computing resources allocated to task i by the LEO satellite's MEC server (cycles per second).

When $x_{i,\mathrm{B}} = 1$, the delay includes the delay that UE i offloads task i to the LEO satellite, the propagation delay between UE i and the LEO satellite, the delay that the LEO satellite migrates task i to the gateway, the propagation delay between the LEO satellite and the gateway, and the delay that the BS's MEC server computes task i. Because there is a wired connection between the gateway and the BS, the corresponding delay between them is very low, so it is ignored here.

$$D_{i,\mathrm{B}} = \frac{a_i}{r_{i,\mathrm{S}}} + \frac{2d_{i,\mathrm{S}}}{c} + \frac{a_i}{r_{\mathrm{S,G}}} + \frac{2d_{\mathrm{S,G}}}{c} + \frac{c_i a_i}{f_{i,\mathrm{B}}}, \tag{5}$$

$f_{i,\mathrm{B}}$ are computing resources allocated to task i by the BS's MEC server (cycles per second).

4. Problem Formulation

The task completion delay of UE i can be expressed as

$$\begin{aligned} U_i &= x_{i,0} D_{i,0} + x_{i,\mathrm{S}} D_{i,\mathrm{S}} + x_{i,\mathrm{B}} D_{i,\mathrm{B}} \\ &= x_{i,0} \frac{c_i a_i}{f_{i,0}} + x_{i,\mathrm{S}} \left(\frac{a_i}{r_{i,\mathrm{S}}} + \frac{2d_{i,\mathrm{S}}}{c} + \frac{c_i a_i}{f_{i,\mathrm{S}}} \right) + x_{i,\mathrm{B}} \left(\frac{a_i}{r_{i,\mathrm{S}}} + \frac{2d_{i,\mathrm{S}}}{c} + \frac{a_i}{r_{\mathrm{S,G}}} + \frac{2d_{\mathrm{S,G}}}{c} + \frac{c_i a_i}{f_{i,\mathrm{B}}} \right). \end{aligned} \tag{6}$$

The completion delay of all UEs' tasks can be expressed as $U = \sum_{i \in \mathcal{N}} U_i$. The optimization objective of this paper is to minimize the completion delay of all UEs' tasks. The optimization variables are the task offloading decisions of UEs, i.e., $x_{i,0}$, $x_{i,\mathrm{S}}$, $x_{i,\mathrm{B}}$ and the computing resource allocation of the LEO satellite's MEC server and the BS's MEC server, i.e., $f_{i,\mathrm{S}}$, $f_{i,\mathrm{B}}$. The optimization problem $\mathcal{P}$ is as follows.

$$\min_{\left\{ \substack{x_{i,0}, x_{i,\mathrm{S}}, x_{i,\mathrm{B}}, \\ f_{i,\mathrm{S}}, f_{i,\mathrm{B}}} \right\}} \quad U \tag{7a}$$

$$\text{s.t.} \sum_{i \in \mathcal{N}} x_{i,\mathrm{S}} f_{i,\mathrm{S}} \le F_{\mathrm{S}}, \tag{7b}$$

$$\sum_{i \in \mathcal{N}} x_{i,\mathrm{B}} f_{i,\mathrm{B}} \le F_{\mathrm{B}}, \tag{7c}$$

$$0 \le f_{i,\mathrm{S}} \le F_{\mathrm{S}}, \forall i \in \mathcal{N}, \tag{7d}$$

$$0 \le f_{i,\mathrm{B}} \le F_{\mathrm{B}}, \forall i \in \mathcal{N}, \tag{7e}$$

$$x_{i,0} + x_{i,\mathrm{S}} + x_{i,\mathrm{B}} = 1, \forall i \in \mathcal{N}, \tag{7f}$$

$$x_{i,0}, x_{i,\mathrm{S}}, x_{i,\mathrm{B}} \in \{0, 1\}, \forall i \in \mathcal{N}. \tag{7g}$$

Equation (7a) is the objective function. (7b) represents that computing resources allocated to task i cannot surpass the resources owned by the LEO satellite's MEC server, i.e., F_{S}. (7c) shows that computing resources allocated to task i cannot exceed the resources owned by the BS's MEC server, i.e., F_{B}. (7d) and (7e) mean that the MEC server's computing resource allocation for task i is nonnegative. (7f) denotes that only one strategy is selected to complete task i. (7g) means $x_{i,0}$, $x_{i,\mathrm{S}}$, and $x_{i,\mathrm{B}}$ are 0 or 1.

5. Algorithm Design

Since $\mathcal{P}$ involves discrete variables, i.e., $x_{i,0}$, $x_{i,\mathrm{S}}$, $x_{i,\mathrm{B}}$, and continuous variables, i.e., $f_{i,\mathrm{S}}$, $f_{i,\mathrm{B}}$, it is a mixed-integer nonlinear programming problem (MINLP), which is NP hard and difficult to solve. Hence, we propose a joint offloading decision and resource allocation scheme based on potential game and Lagrange multiplier method, where $\mathcal{P}$ is decomposed into two sub-problems, i.e., the LEO satellite's and the BS's computing resource allocation, and UEs' task offloading decision.

5.1. Resource Allocation

Assuming that $x^*_{i,0}$, $x^*_{i,\mathrm{S}}$, and $x^*_{i,\mathrm{B}}$ are fixed, the sets $\mathcal{N}^D = \{i|x^*_{i,0} = 1\}$, $\mathcal{N}^\mathrm{S} = \{i|x^*_{i,\mathrm{S}} = 1\}$, and $\mathcal{N}^\mathrm{B} = \{i|x^*_{i,\mathrm{B}} = 1\}$ can be obtained, $\mathcal{P}$ can be transformed into the problem $\mathcal{P}1$.

$$\min_{\{f_{i,\mathrm{S}}, f_{i,\mathrm{B}}\}} \sum_{i\in\mathcal{N}^\mathrm{D}} \frac{c_i a_i}{f_{i,0}} + \sum_{i\in\mathcal{N}^\mathrm{S}} \left(\frac{a_i}{r_{i,\mathrm{S}}} + \frac{2d_{i,\mathrm{S}}}{c} + \frac{c_i a_i}{f_{i,\mathrm{S}}}\right) + \sum_{i\in\mathcal{N}^\mathrm{B}} \left(\frac{a_i}{r_{i,\mathrm{S}}} + \frac{2d_{i,\mathrm{S}}}{c} + \frac{a_i}{r_{\mathrm{S,G}}} + \frac{2d_{\mathrm{S,G}}}{c} + \frac{c_i a_i}{f_{i,\mathrm{B}}}\right) \tag{8a}$$

$$\text{s.t.} \sum_{i\in\mathcal{N}^\mathrm{S}} f_{i,\mathrm{S}} \le F_\mathrm{S}, \tag{8b}$$

$$\sum_{i\in\mathcal{N}^\mathrm{B}} f_{i,\mathrm{B}} \le F_\mathrm{B}, \tag{8c}$$

$$0 \le f_{i,\mathrm{S}} \le F_\mathrm{S}, \forall i \in \mathcal{N}^\mathrm{S}, \tag{8d}$$

$$0 \le f_{i,\mathrm{B}} \le F_\mathrm{B}, \forall i \in \mathcal{N}^\mathrm{B}. \tag{8e}$$

Equation (8a) is the objective function. The meanings of (8b)–(8e) are similar with those of (7b)–(7e). It can be seen from $\mathcal{P}1$ that the local computing delay is a constant, and the computing resource allocation of the LEO satellite and that of the BS are independent. Therefore, $\mathcal{P}1$ can be decomposed into two sub-problems, i.e., $\mathcal{P}1.1$ and $\mathcal{P}1.2$, then $f^*_{i,\mathrm{S}}$ and $f^*_{i,\mathrm{B}}$ can be obtained by solving them separately.

$\mathcal{P}1.1$ is the computing resource allocation of the LEO satellite's MEC server, which can be expressed as

$$\min_{\{f_{i,\mathrm{S}}\}} \sum_{i\in\mathcal{N}^\mathrm{S}} \left(\frac{a_i}{r_{i,\mathrm{S}}} + \frac{2d_{i,\mathrm{S}}}{c} + \frac{c_i a_i}{f_{i,\mathrm{S}}}\right) \quad \text{s.t. (8b) and (8d).} \tag{9}$$

The sum in (9) is denoted by U^S. $\partial U^\mathrm{S}/\partial f_{i,\mathrm{S}} = -c_i a_i/(f_{i,\mathrm{S}})^2$. The first derivative is less than 0. $\partial^2 U^\mathrm{S}/\partial(f_{i,\mathrm{S}})^2 = 2c_i a_i/(f_{i,\mathrm{S}})^3$. The second derivative is greater than 0. $\partial^2 U^\mathrm{S}/(\partial f_{i,\mathrm{S}} \partial f_{j,\mathrm{S}}) = 0$, $i, j \in \mathcal{N}^\mathrm{S}$, $i \neq j$. The second mixed partial derivative is equal to 0. Therefore, the Hessian matrix of U^S is positive definite, and U^S is convex. $\mathcal{P}1.1$ can be solved by using the Lagrange multiplier method, the Lagrange function is constructed as

$$L(f_{i,\mathrm{S}}, \theta) = \sum_{i\in\mathcal{N}^\mathrm{S}} \left(\frac{a_i}{r_{i,\mathrm{S}}} + \frac{2d_{i,\mathrm{S}}}{c} + \frac{c_i a_i}{f_{i,\mathrm{S}}}\right) + \theta\left(\sum_{i\in\mathcal{N}^\mathrm{S}} f_{i,\mathrm{S}} - F_\mathrm{S}\right), \tag{10}$$

where θ is the Lagrange multiplier and $\theta \ge 0$. Karush–Kuhn–Tucker (KKT) conditions are listed as

$$\frac{\partial L(f_{i,\mathrm{S}}, \theta)}{\partial f_{i,\mathrm{S}}} = -\frac{c_i a_i}{(f_{i,\mathrm{S}})^2} + \theta = 0, \tag{11}$$

$$\frac{\partial L(f_{i,\mathrm{S}}, \theta)}{\partial \theta} = \sum_{i\in\mathcal{N}^\mathrm{S}} f_{i,\mathrm{S}} - F_\mathrm{S} = 0, \tag{12}$$

$$\theta\left(\sum_{i\in\mathcal{N}^{\mathrm{S}}} f_{i,\mathrm{S}} - F_{\mathrm{S}}\right) = 0, \tag{13}$$

$$\theta \geq 0, \tag{14}$$

$$\text{(8b) and (8d).}$$

$f_{i,\mathrm{S}} = \sqrt{c_i a_i/\theta}$ is obtained from (11). Substituting it into (12), we can get $\sum_{i\in\mathcal{N}^{\mathrm{S}}} \sqrt{c_i a_i/\theta} - F_{\mathrm{S}} = 0$, i.e., $\sum_{i\in\mathcal{N}^{\mathrm{S}}} \sqrt{c_i a_i} = F_{\mathrm{S}}\sqrt{\theta}$. So $f_{i,\mathrm{S}} = \sqrt{c_i a_i}/\sqrt{\theta} = F_{\mathrm{S}}\sqrt{c_i a_i}/(\sum_{i\in\mathcal{N}^{\mathrm{S}}} \sqrt{c_i a_i})$. Considering (8d), we can deduce

$$f_{i,\mathrm{S}}^{*} = \min\left[\max\left(\frac{F_{\mathrm{S}}\sqrt{c_i a_i}}{\sum\limits_{i\in\mathcal{N}^{\mathrm{S}}}\sqrt{c_i a_i}}, 0\right), F_{\mathrm{S}}\right]. \tag{15}$$

$\mathcal{P}1.2$ is the computing resource allocation of the BS's MEC server, which can be expressed as

$$\min_{\{f_{i,\mathrm{B}}\}} \sum_{i\in\mathcal{N}^{\mathrm{B}}} \left(\frac{a_i}{r_{i,\mathrm{S}}} + \frac{2d_{i,\mathrm{S}}}{c} + \frac{a_i}{r_{\mathrm{S,G}}} + \frac{2d_{\mathrm{S,G}}}{c} + \frac{c_i a_i}{f_{i,\mathrm{B}}}\right) \tag{16}$$
$$\text{s.t. (8c) and (8e).}$$

The sum in (16) is denoted by U^{B}. $\partial U^{\mathrm{B}}/\partial f_{i,\mathrm{B}} = -c_i a_i/(f_{i,\mathrm{B}})^2$. The first derivative is less than 0. $\partial^2 U^{\mathrm{B}}/\partial (f_{i,\mathrm{B}})^2 = 2c_i a_i/(f_{i,\mathrm{B}})^3$. The second derivative is greater than 0. $\partial^2 U^{\mathrm{B}}/(\partial f_{i,\mathrm{B}} \partial f_{j,\mathrm{B}}) = 0$, $i, j \in \mathcal{N}^{\mathrm{B}}$, $i \neq j$. The second mixed partial derivative is equal to 0. Therefore, the Hessian matrix of U^{B} is positive definite, and U^{B} is convex. $\mathcal{P}1.2$ can be solved by using the Lagrange multiplier method, the Lagrange function is constructed as

$$L(f_{i,\mathrm{B}}, \lambda) = \sum_{i\in\mathcal{N}^{\mathrm{B}}} \left(\frac{a_i}{r_{i,\mathrm{S}}} + \frac{2d_{i,\mathrm{S}}}{c} + \frac{a_i}{r_{\mathrm{S,G}}} + \frac{2d_{\mathrm{S,G}}}{c} + \frac{c_i a_i}{f_{i,\mathrm{B}}}\right) + \lambda\left(\sum_{i\in\mathcal{N}^{\mathrm{B}}} f_{i,\mathrm{B}} - F_{\mathrm{B}}\right), \tag{17}$$

where λ is the Lagrange multiplier and $\lambda \geq 0$. KKT conditions are listed as

$$\frac{\partial L(f_{i,\mathrm{B}}, \lambda)}{\partial f_{i,\mathrm{B}}} = -\frac{c_i a_i}{(f_{i,\mathrm{B}})^2} + \lambda = 0, \tag{18}$$

$$\frac{\partial L(f_{i,\mathrm{B}}, \lambda)}{\partial \lambda} = \sum_{i\in\mathcal{N}^{\mathrm{B}}} f_{i,\mathrm{B}} - F_{\mathrm{B}} = 0, \tag{19}$$

$$\lambda\left(\sum_{i\in\mathcal{N}^{\mathrm{B}}} f_{i,\mathrm{B}} - F_{\mathrm{B}}\right) = 0, \tag{20}$$

$$\lambda \geq 0, \tag{21}$$

$$\text{(8c) and (8e).}$$

It is similar to the solution process of $\mathcal{P}1.1$, $f_{i,\mathrm{B}} = \sqrt{c_i a_i/\lambda}$ is obtained from (18). Substituting it into (19), we can get $\sum_{i\in\mathcal{N}^{\mathrm{B}}} \sqrt{c_i a_i/\lambda} - F_{\mathrm{B}} = 0$, i.e., $\sum_{i\in\mathcal{N}^{\mathrm{B}}} \sqrt{c_i a_i} = F_{\mathrm{B}}\sqrt{\lambda}$. So $f_{i,\mathrm{B}} = \sqrt{c_i a_i}/\sqrt{\lambda} = F_{\mathrm{B}}\sqrt{c_i a_i}/(\sum_{i\in\mathcal{N}^{\mathrm{B}}} \sqrt{c_i a_i})$. Considering (8e), we can deduce

$$f_{i,\mathrm{B}}^{*} = \min\left[\max\left(\frac{F_{\mathrm{B}}\sqrt{c_i a_i}}{\sum\limits_{i\in\mathcal{N}^{\mathrm{B}}}\sqrt{c_i a_i}}, 0\right), F_{\mathrm{B}}\right]. \tag{22}$$

5.2. Offloading Decision

When $f_{i,\mathrm{S}}^{*}$ and $f_{i,\mathrm{B}}^{*}$ are given, $\mathcal{P}$ can be reformulated to the problem $\mathcal{P}2$.

$$\min_{\{x_{i,0},x_{i,\mathrm{S}},x_{i,\mathrm{B}}\}} \sum_{i\in\mathcal{N}} \left[x_{i,0}\frac{c_i a_i}{f_{i,0}} + x_{i,\mathrm{S}}\left(\frac{a_i}{r_{i,\mathrm{S}}} + \frac{2d_{i,\mathrm{S}}}{c} + \frac{c_i a_i}{f_{i,\mathrm{S}}}\right) + x_{i,\mathrm{B}}\left(\frac{a_i}{r_{i,\mathrm{S}}} + \frac{2d_{i,\mathrm{S}}}{c} + \frac{a_i}{r_{\mathrm{S,G}}} + \frac{2d_{\mathrm{S,G}}}{c} + \frac{c_i a_i}{f_{i,\mathrm{B}}}\right)\right] \tag{23}$$

s.t. (7f) and (7g).

This is an integer programming (IP) problem, which is difficult to solve by traditional optimization methods, e.g., the convex optimization [30]. Because computing resources of the LEO satellite's MEC server and the BS's MEC server are limited, there is a fierce competition for them among UEs. Therefore, this paper considers $\mathcal{P}2$ as a multi-UEs task offloading game and uses the game theory approach to solve it, so as to obtain an approximate optimal solution within acceptable time.

$\mathcal{P}2$ can be constructed as a game $\mathcal{G} = \{\mathcal{N}, \mathcal{Y}_i, V_i\}$, where $\mathcal{N}$ is the set of participants, i.e., UEs. $\mathcal{Y}_i = \{0,1,2\}$ is the strategy space, i.e., the set of all optional task offloading strategies of UE i, and y_i represents the task offloading decision of UE i. $y_i = 0$ denotes $\{x_{i,0} = 1, x_{i,\mathrm{S}} = 0, x_{i,\mathrm{B}} = 0\}$, $y_i = 1$ denotes $\{x_{i,0} = 0, x_{i,\mathrm{S}} = 1, x_{i,\mathrm{B}} = 0\}$, $y_i = 2$ denotes $\{x_{i,0} = 0, x_{i,\mathrm{S}} = 0, x_{i,\mathrm{B}} = 1\}$. V_i is the payoff function of UE i, because the payoff needs to be maximized, and U_i should be minimized, so $V_i = -U_i$. $V_i(y_i, y_{-i})$ represents the payoff that UE i selects y_i when y_{-i} is given, where y_{-i} indicates the task offloading decisions of UEs except UE i. In the game process, each UE is purely rational or selfish, hoping to make the task offloading decision that can obtain the most payoff. The decision of each UE affects the payoffs of other UEs, so it is necessary to obtain an optimal decision set recognized by all UEs [31]. This is the solution of the game process, i.e., NE, which is defined as follows:

Definition 1. *Nash equilibrium. If and only if no UE i can unilaterally change its task offloading decision to increase its own payoff when the task offloading decisions of other UEs remain unchanged, the task offloading decision set* $y^* = (y_1^*, y_2^*, y_3^*, \ldots, y_N^*)$ *is an NE of* $\mathcal{G}$ *and can be expressed as* $V_i(y_i^*, y_{-i}^*) \geq V_i(y_i, y_{-i}^*), \forall i \in \mathcal{N}, \forall y_i^*, y_i \in \mathcal{Y}_i, y_i^* \neq y_i$.

In this case, the game reaches an equilibrium state, and the task offloading decision set can be accepted by all UEs. The EPG is defined as follows:

Definition 2. *Exact potential game. If and only if there exists a potential function* Γ*, which satisfies* $V_i(y_i, y_{-i}) - V_i({y'}_i, y_{-i}) = \Gamma(y_i, y_{-i}) - \Gamma({y'}_i, y_{-i}), \forall i \in \mathcal{N}, \forall y_i, {y'}_i \in \mathcal{Y}_i, y_i \neq {y'}_i$. *The game is an EPG [32].*

No matter which UE changes its task offloading decision, its payoff function and potential function will show the same variation tendency.

Corollary 1. $\mathcal{G}$ *is an EPG with at least one pure strategy NE and finite improvement property (FIP).*

Proof of Corollary 1. The potential function is set to be $\Gamma(y_i, y_{-i}) = -\sum_{i\in\mathcal{N}} U_i(y_i, y_{-i})$. When UE i changes y_i to ${y'}_i$. The variation of its payoff function is $V_i(y_i, y_{-i}) - V_i({y'}_i, y_{-i}) = -U_i(y_i, y_{-i}) + U_i({y'}_i, y_{-i})$. The variation of the potential function is $\Gamma(y_i, y_{-i}) - \Gamma({y'}_i, y_{-i}) = -\sum_{i\in\mathcal{N}} U_i(y_i, y_{-i}) + \sum_{i\in\mathcal{N}} U_i({y'}_i, y_{-i}) = -U_i(y_i, y_{-i}) - \sum_{n\neq i} U_n(y_n, y_{-n}) + U_i({y'}_i, y_{-i}) + \sum_{n\neq i} U_n(y_n, y_{-n}) = -U_i(y_i, y_{-i}) + U_i({y'}_i, y_{-i})$. Thus, there exists $V_i(y_i, y_{-i}) - V_i({y'}_i, y_{-i}) = \Gamma(y_i, y_{-i}) - \Gamma({y'}_i, y_{-i}), \forall i \in \mathcal{N}, \forall y_i, {y'}_i \in \mathcal{Y}_i, y_i \neq {y'}_i$. $\mathcal{G}$ is an EPG. The strategy space of $\mathcal{G}$ is finite. According to [32], each finite potential game has a pure strategy NE and FIP. So $\mathcal{G}$ has a pure strategy NE and FIP. Corollary 1 is proved. □

Regardless of UEs' initial strategy selection and change order, $\mathcal{G}$ will converge to an NE within a finite number of iterations that can increase the payoff [33].

5.3. Joint Offloading Decision and Resource Allocation

The joint offloading decision and resource allocation scheme based on potential game, Lagrange multiplier method, and satellite-terrestrial cooperation (JODRA-PGLMC) is proposed in this paper. t is the iteration number, $t^{\max}$ is the maximum iteration number. $y(t)$ denotes UEs' task offloading decisions in the $(t-1)$th iteration. All UEs' task offloading decisions are initialized to local computing. During each iteration, each UE tries to change its task offloading decision while the task offloading decisions of other UEs remain unchanged. If the UE gets more payoff, its task offloading decision will be changed, otherwise, its task offloading decision will be unchanged. The payoff is bounded due to the limited strategy space, when no UE can gain more payoff by changing its task offloading decision, the algorithm will converge and obtain the local or global optimal solution of $\mathcal{P}$ [34]. In this case, the game reaches the NE. $y(t)$ is equivalent to $y^* = \left(y_1^*, y_2^*, y_3^*, \ldots, y_N^*\right)$, which is the optimal task offloading decisions of UEs, i.e., $x_{i,0}^*$, $x_{i,\mathrm{S}}^*$, $x_{i,\mathrm{B}}^*$, corresponding $f_{i,\mathrm{S}}^*$ and $f_{i,\mathrm{B}}^*$ are the optimal computing resource allocation. Thus, the minimum U can be obtained. The complexity of the algorithm is $\mathcal{O}(t^{\max}N)$. The detailed steps of JODRA-PGLMC is expressed in Algorithm 1.

Algorithm 1 JODRA-PGLMC

Initialize $y(1) = \{y_i = 0, \forall i \in \mathcal{N}\}$. Set $t = 1$. Set $t^{\max}$.
while $1 \le t \le t^{\max}$ **do**
  $t = t + 1$.
  for each UE $i \in \mathcal{N}$ **do**
    Calculate $V_i(y_i(t-1), y_{-i}(t-1))$.
    Obtain y'_i and y''_i, $y'_i, y''_i \in \mathcal{Y}_i / \{y_i(t-1)\}$, $y'_i \neq y''_i$.
    if $y'_i = 0$ or $y''_i = 0$ **then**
      calculate $V_i(y'_i, y_{-i}(t-1))$ or $V_i(y''_i, y_{-i}(t-1))$.
    else if $y'_i = 1$ or $y''_i = 1$ **then**
      obtain $f_{i,\mathrm{S}}^*$ by (15), so as to calculate $V_i(y'_i, y_{-i}(t-1))$ or $V_i(y''_i, y_{-i}(t-1))$.
    else if $y'_i = 2$ or $y''_i = 2$ **then**
      obtain $f_{i,\mathrm{B}}^*$ by (22), so as to calculate $V_i(y'_i, y_{-i}(t-1))$ or $V_i(y''_i, y_{-i}(t-1))$.
    end if
    Compare $V_i(y_i(t-1), y_{-i}(t-1))$, $V_i(y'_i, y_{-i}(t-1))$ with $V_i(y''_i, y_{-i}(t-1))$, update $y_i(t-1)$ to $y_i(t)$ using the task offload decision corresponding to the maximum among them.
  end for
  if $y(t) = y(t-1)$ **then**
    break.
  end if
end while
return $y(t)$.

6. Results and Discussion

This section lists the simulation parameters. The optimization objective is evaluated by comparing our scheme with other baseline schemes. The simulation results are presented and discussed.

6.1. Simulation Parameter

Except for the variable parameter of each simulation, the simulation parameters are shown in Tables 1 and 2. a_i, $f_{i,0}$, θ_{u}, and θ_{g} obey the random distribution. K is the Ricean factor, which is associated with θ_{u}, θ_{g} and obeys the normal distribution, μ_{K} is the mean, σ_{K} is the standard deviation. The main references for parameter setting are [11,17,21,26,27]. To validate JODRA-PGLMC, it is compared with other baseline schemes. The baseline schemes are as follows. The joint offloading decision and resource allocation scheme based on random decision, Lagrange multiplier method, and satellite-terrestrial cooperation

(JODRA-RDLMC). The joint offloading decision and resource allocation scheme based on potential game, Lagrange multiplier method, and non-cooperation (JODRA-PGLMNC). The scheme that only considers computation offloading, others are the same as our scheme (OCO). The scheme only considers local computing (OLC). All simulation results are obtained by running MATLAB for 2000 times and taking the average value.

Table 1. Simulation parameters.

Parameter	Value	Parameter	Value
N	40	p_i	2 W
c_i	1000 cycles/bit	a_i	[0.5, 5] Mbit
$f_{i,0}$	[1, 2] GHz	g_i	43.2 dBi
$B_{i,\mathrm{S}}$	800 MHz	$B_{\mathrm{S,G}}$	800 MHz
f_u	30 GHz	f_d	20 GHz
$N_{0_{i,\mathrm{S}}}$	−203 dBm/Hz	$N_{0_{\mathrm{S,G}}}$	−203 dBm/Hz
h_s	780 km	P	50 W
D_TA	0.5 m	D_RA	0.33 m
F_S	15 Gcycles/s	F_B	60 Gcycles/s
D_g	7.3 m	c	3×10^8 m/s
R_e	6371 km	$t^{\max}$	2000
θ_u	[10°, 90°]	θ_g	[10°, 30°]

Table 2. Simulation parameters.

Parameter	Value	Value	Value	Value	Value	Value	Value	Value	Value
$\theta_\mathrm{u}, \theta_\mathrm{g}$	10°	20°	30°	40°	50°	60°	70°	80°	90°
μ_K (dB)	25.43	12.72	8.40	6.52	5.24	4.57	4.02	3.70	3.62
σ_K (dB)	7.04	7.47	7.18	6.88	5.28	4.92	3.40	2.22	2.28

6.2. Simulation Results and Disscusion

As depicted in Figure 3, as the number of UEs, i.e., N increases, the delay of all schemes increases. The delay of our scheme is always minimal. When $N \leq 30$, the delay of JODRA-PGLMC and that of OCO are almost the same. When $N > 30$, the gap becomes apparent gradually. It can be seen that there is an intersection for OCO and OLC between $N = 50$ and $N = 60$. When $N \leq 50$, JODRA-PGLMC and OCO are closer. When $N \geq 60$, JODRA-PGLMC and OLC are closer. This indicates that when N is small, UEs can obtain more computing resources by computation offloading. Therefore, most UEs tend to offload tasks to the LEO satellite or the BS. When N is large, UEs can obtain few computing resources from the LEO satellite or the BS. So UEs tend to compute tasks locally. The delay of JODRA-PGLMC is lower than that of JODRA-RDLMC, and the gap increases gradually, indicating that potential game can obtain a better task offloading decision effectively. The difference between JODRA-PGLMC and JODRA-PGLMNC means the gain brought by the satellite-terrestrial cooperation. The satellite migrates tasks to the BS in an adjacent area can bring stable gain. Comparing with the difference between JODRA-PGLMC and OLC, the difference between JODRA-PGLMNC and OLC is smaller, because computing resources of the LEO satellite's MEC server are much less than those of the BS's MEC server. The gain that can be brought by offloading tasks to the LEO satellite is limited. The delay of JODRA-PGLMC is always lower than that of OLC, and they have a relatively stable difference, which demonstrates that no matter offloading tasks to the LEO satellite or the BS can bring gain for UEs, and also confirms the effectiveness of the proposed scheme.

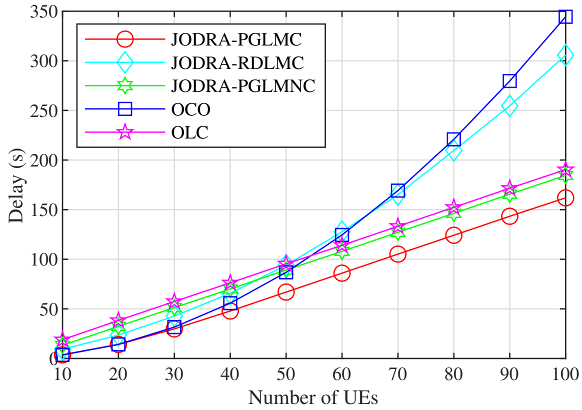

Figure 3. Delay vs. number of UEs.

As shown in Figure 4, with the increase of computation intensity, i.e., c_i, the delay of all schemes increases. Because computing resources required by tasks increase, the corresponding task computing delay also increases. The delay of our scheme is lower than that of JODRA-RDLMC, which indicates that the potential game can obtain an appropriate task offloading decision, thus reducing the delay. Comparing with the delay of JODRA-PGLMNC, the delay of JODRA-PGLMC is lower, and the difference between them increases gradually, which means that offloading tasks to the BS can bring higher gain with the increase of c_i, and also illustrates the necessity of satellite-terrestrial cooperation. The delay of JODRA-PGLMC is lower than that of OCO, and they are the closest. The reason is that the more computing resources tasks require, the fewer UEs that the LEO satellite can assist, many tasks are migrated to the BS for computing. The delay of OLC is the highest among all schemes, and the difference between OLC and JODRA-PGLMC is getting bigger and bigger. This proves that both the proper task offloading decision and the proper computing resource allocation are very important and can bring significant gain for UEs.

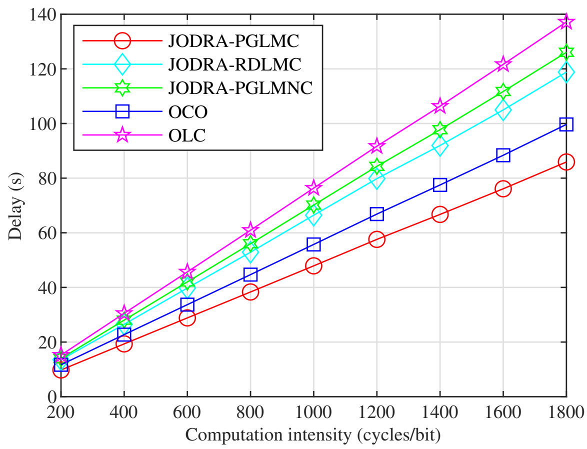

Figure 4. Delay vs. computation intensity.

As depicted in Figure 5, it is assumed that the data size of each task, i.e., a_i is the same, and the delay of all schemes increases with the increase of a_i. Because bigger a_i will lead to higher task transmission delay and task migration delay. Moreover, when c_i is fixed, computing resources required by tasks are proportional to a_i. In other words, task computing delay will also increase with the increase of a_i. The delay of JODRA-PGLMC is much lower than that of JODRA-RDLMC, and the difference increases gradually, which indicates that potential game can make UEs' task offloading decisions effectively. The difference between JODRA-PGLMC's delay and JODRA-PGLMNC's delay represents the gain brought by migrating task from the satellite to the BS for computing. This gradually increasing difference proves the effectiveness and necessity of satellite-terrestrial cooperation. The delay of JODRA-PGLMC is closest to that of OCO, and is furthest from that of OLC. This means that when N is small, e.g., $N = 40$, computation offloading can obtain more gain. The delay of JODRA-PGLMC is always the lowest among all schemes, which indicates that the proposed scheme can reduce UEs' task completion delay successfully.

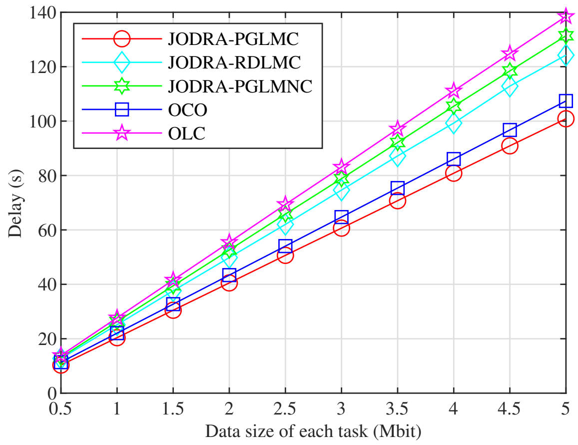

Figure 5. Delay vs. data size of each task.

As shown in Figure 6, with the increase of computing resources of the satellite, i.e., F_S, the delay of all schemes except OLC presents a decreasing trend. In addition, the delay of JODRA-PGLMC, JODRA-PGLMNC, and OCO decreases slowly, while that of JODRA-RDLMC decreases rapidly. Because the more computing resources the satellite has, the more computing resources can be allocated to UEs' tasks, which reduces the task computing delay. However, computing resources of the satellite's MEC server are far less than those of the BS's MEC server, the gain brought for UEs is limited. Therefore, the number of UEs that offload tasks to the satellite for computing is relatively small. In JODRA-RDLMC, each UE is likely to offload its task to the satellite for computing, i.e., more tasks are computed by the satellite's MEC server. Therefore, its delay is affected by F_S significantly. However, because the random decision is not optimal, its delay is high, even the highest when $F_S < 12$ Gcycles/s. The delay of JODRA-PGLMC is lower than that of JODRA-RDLMC, indicating that a suitable task offloading decision can be obtained by potential game. Compared with JODRA-PGLMNC's delay, JODRA-PGLMC's delay is lower, and the difference between them is relatively stable, which illustrates that satellite-terrestrial cooperation is effective. The delay of OCO is higher than that of JODRA-PGLMC, which means that all UEs offload tasks to the MEC server is not optimal, and also indicates the importance of an appropriate task offloading decision. Because OLC is independent of computation offloading, its delay is not affected by changes of F_S, and remains unchanged basically. The great difference between its delay and JODRA-PGLMC's delay proves the effectiveness of the proposed scheme.

As depicted in Figure 7, with the increase of computing resources of the BS, i.e., F_B, the delay of JODRA-PGLMNC and OLC remains unchanged basically, while the delay of other schemes decreases gradually. Since JODRA-PGLMNC and OLC do not involve the BS, their delay is not affected by the variation of F_B. The more computing resources the BS has, the more computing resources it can allocate to UEs' tasks, thus reducing task computing delay. There is an intersection between JODRA-RDLMC and OCO. When $F_B \leq 45$ Gcycles/s, the delay of JODRA-PGLMC is closer to that of JODRA-RDLMC. When $F_B > 45$ Gcycles/s, the delay of JODRA-PGLMC is closer to that of OCO. The reason is that when F_B is few, the gain from computation offloading is relatively limited. If a UE makes a task offloading decision randomly, local computing may be selected, which can obtain more gain. When F_B is many, computation offloading can provide more gain for UEs. It can be seen that when F_B is abundant, the delay of OCO is very close to that of JODRA-PGLMC. This indicates that rich computing resources of the BS make more UEs tend to offload tasks. The delay of

JODRA-PGLMC is much lower than that of JODRA-RDLMC, and the difference increases gradually, which proves that potential game is effective in making the task offloading decision. Comparing with JODRA-PGLMNC's delay, JODRA-PGLMC's delay is lower, and the difference increases with the increase of F_B, which is due to the satellite-terrestrial cooperation. F_B affects the gain for UEs provided by satellite-terrestrial cooperation directly. The difference between JODRA-PGLMC's delay and OLC's delay is almost always the largest, which proves that the proposed scheme is quite beneficial to UEs.

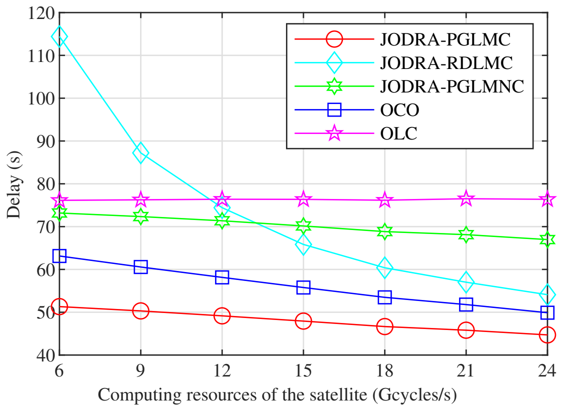

Figure 6. Delay vs. computing resources of the satellite.

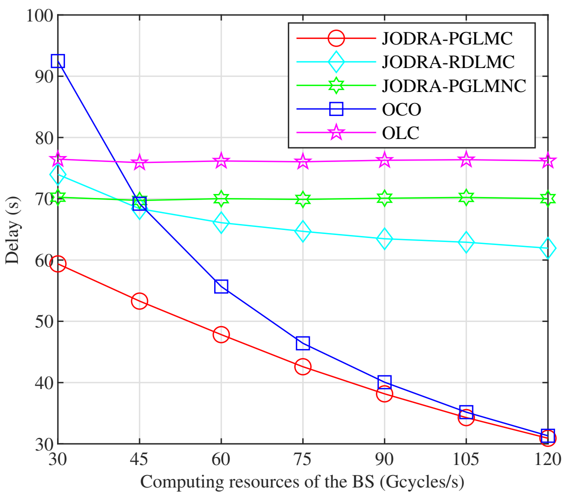

Figure 7. Delay vs. computing resources of the BS.

7. Conclusions

In this paper, the symmetry between satellite networks and terrestrial networks, and the asymmetry between their resources are considered. We research the satellite-terrestrial cooperation in MEC-enabled STN. Different from a bent pipe satellite and a regenerative satellite, we take into account a satellite, which functions as both a relay and a computing unit to assist UEs in computing tasks. Most existing researches ignore on-board processing capabilities of satellites, whereas this paper makes full and effective use of the capabilities. We design a system model of MEC-enabled STN. Because tasks are indivisible, computation can be performed at the UE, the satellite, or the BS. Aiming to minimize all UEs' task completion delay, the corresponding optimization problem is formulated, and then decomposed into a task offloading decision problem and a computing resource allocation problem. A joint offloading decision and resource allocation scheme based on potential game and Lagrange multiplier method is proposed to solve these problems. Simulation results show that compared with other baseline schemes, the proposed scheme can obtain lower completion delay of all UEs' tasks. Although on-board processing capabilities of satellites are currently limited, with the continuous development of hardware and software technology, the capabilities will be improved gradually. The research of this paper is a theoretical exploration of a new research direction, which may be a reference for researchers in this research direction or related research directions. If the research contributes in any way to academia and industry, we will be honored and happy. There is still work to be done in the future, e.g., considering inter-satellite cooperation, which uses computing resources of MEC servers deployed at adjacent satellites to compute tasks.

Author Contributions: Conceptualization, M.T.; Data curation, M.T.; Formal analysis, M.T.; Funding acquisition, X.W.; Investigation, M.T.; Methodology, M.T. and S.L.; Project administration, X.W. and S.L.; Resources, X.W.; Software, M.T.; Supervision, S.L.; Validation, M.T.; Visualization, M.T.; Writing—original draft preparation, M.T.; writing—review and editing, M.T., X.W., S.L. and L.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Program of China grant number 2019YFB1406500.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank the editors and the reviewers for their helpful suggestions and constructive comments.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhang, L.; Chai, R.; Yang, T.; Chen, Q. Min-Max Worst-Case Design for Computation Offloading in Multi-user MEC System. In Proceedings of the IEEE INFOCOM 2020—IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Toronto, ON, Canada, 6–9 July 2020; pp. 1075–1080. [CrossRef]
2. Liu, J.; Zhang, Q. Computation Resource Allocation for Heterogeneous Time-Critical IoT Services in MEC. In Proceedings of the 2020 IEEE Wireless Communications and Networking Conference (WCNC), Seoul, Korea, 25–28 May 2020; pp. 1–6. [CrossRef]
3. Huang, M.; Liu, W.; Wang, T.; Liu, A.; Zhang, S. A Cloud–MEC Collaborative Task Offloading Scheme With Service Orchestration. *IEEE Internet Things J.* **2020**, *7*, 5792–5805. [CrossRef]
4. Ma, Y.; Liang, W.; Huang, M.; Xu, W.; Guo, S. Virtual Network Function Service Provisioning in MEC via Trading Off the Usages between Computing and Communication Resources. *IEEE Trans. Cloud Comput.* 2020, *in press*. [CrossRef]
5. Li, D.; Jin, Y.; Liu, H. Resource Allocation Strategy of Edge Systems Based on Task Priority and an Optimal Integer Linear Programming Algorithm. *Symmetry* **2020**, *12*, 972. [CrossRef]
6. Berno, M.; Alcaraz, J.J.; Rossi, M. On the Allocation of Computing Tasks under QoS Constraints in Hierarchical MEC Architectures. In Proceedings of the 2019 Fourth International Conference on Fog and Mobile Edge Computing (FMEC), Rome, Italy, 10–13 June 2019; pp. 37–44. [CrossRef]
7. Chen, Y.; Zhang, N.; Zhang, Y.; Chen, X.; Wu, W.; Shen, X. Energy Efficient Dynamic Offloading in Mobile Edge Computing for Internet of Things. *IEEE Trans. Cloud Comput.* **2021**, *9*, 1050–1060. [CrossRef]
8. Chen, S.; Liang, Y.C.; Sun, S.; Kang, S.; Cheng, W.; Peng, M. Vision, Requirements, and Technology Trend of 6G: How to Tackle the Challenges of System Coverage, Capacity, User Data-Rate and Movement Speed. *IEEE Wirel. Commun.* **2020**, *27*, 218–228. [CrossRef]
9. Zhang, Y.; Tang, Y.; Wang, W. Service Deployment and Service Request Optimization Scheduling in MEC enabled LEO Networks. In Proceedings of the 2021 International Conference on Computer Communications and Networks (ICCCN), Athens, Greece, 19–22 July 2021; pp. 1–6. [CrossRef]
10. Wang, P.; Zhang, J.; Zhang, X.; Yan, Z.; Evans, B.G.; Wang, W. Convergence of Satellite and Terrestrial Networks: A Comprehensive Survey. *IEEE Access* **2020**, *8*, 5550–5588. [CrossRef]
11. Deng, R.; Di, B.; Chen, S.; Sun, S.; Song, L. Ultra-Dense LEO Satellite Offloading for Terrestrial Networks: How Much to Pay the Satellite Operator? *IEEE Trans. Wirel. Commun.* **2020**, *19*, 6240–6254. [CrossRef]
12. Wang, B.; Feng, T.; Huang, D. A Joint Computation Offloading and Resource Allocation Strategy for LEO Satellite Edge Computing System. In Proceedings of the 2020 IEEE 20th International Conference on Communication Technology (ICCT), Nanning, China, 28–31 October 2020; pp. 649–655. [CrossRef]
13. Li, C.; Zhang, Y.; Hao, X.; Huang, T. Jointly optimized request dispatching and service placement for MEC in LEO network. *China Commun.* **2020**, *17*, 199–208. [CrossRef]
14. Tang, Q.Q.; Xie, R.C.; Liu, X.; Zhang, Y.S.; He, C.; Li, C.C.; Huang, T. MEC enabled satellite-terrestrial network: Architecture, key technique and challenge. *J. Commun.* **2020**, *41*, 162–181. (In Chinese) [CrossRef]

15. Li, Z.; Li, M.; Wang, Q. Predator–Prey Model Based Asymmetry Resource Allocation in Satellite–Terrestrial Network. *Symmetry* **2021**, *13*, 2113. [CrossRef]
16. Feng, W.; Liu, H.; Yao, Y.; Cao, D.; Zhao, M. Latency-Aware Offloading for Mobile Edge Computing Networks. *IEEE Commun. Lett.* **2021**, *25*, 2673–2677. [CrossRef]
17. Wang, Y.; Zhang, J.; Zhang, X.; Wang, P.; Liu, L. A Computation Offloading Strategy in Satellite Terrestrial Networks with Double Edge Computing. In Proceedings of the 2018 IEEE International Conference on Communication Systems (ICCS), Chengdu, China, 19–21 December 2018; pp. 450–455. [CrossRef]
18. Zhang, Z.; Zhang, W.; Tseng, F.H. Satellite Mobile Edge Computing: Improving QoS of High-Speed Satellite-Terrestrial Networks Using Edge Computing Techniques. *IEEE Netw.* **2019**, *33*, 70–76. [CrossRef]
19. Cheng, N.; Lyu, F.; Quan, W.; Zhou, C.; He, H.; Shi, W. Space/Aerial-Assisted Computing Offloading for IoT Applications: A Learning-Based Approach. *IEEE J. Sel. Areas Commun.* **2019**, *37*, 1117–1129. [CrossRef]
20. Xie, R.; Tang, Q.; Wang, Q.; Liu, X.; Yu, F.R.; Huang, T. Satellite-Terrestrial Integrated Edge Computing Networks: Architecture, Challenges, and Open Issues. *IEEE Netw.* **2020**, *34*, 224–231. [CrossRef]
21. Zhang, J.; Zhang, X.; Wang, P.; Liu, L.; Wang, Y. Double-edge intelligent integrated satellite terrestrial networks. *China Commun.* **2020**, *17*, 128–146. [CrossRef]
22. Chen, T.; Liu, J.; Ye, Q.; Zhuang, W.; Zhang, W.; Huang, T.; Liu, Y. Learning-based Computation Offloading for IoRT through Ka/Q-Band Satellite-Terrestrial Integrated Networks. *IEEE Internet Things J.* 2021, *in press*. [CrossRef]
23. Song, Z.; Hao, Y.; Liu, Y.; Sun, X. Energy-Efficient Multiaccess Edge Computing for Terrestrial-Satellite Internet of Things. *IEEE Internet Things J.* **2021**, *8*, 14202–14218. [CrossRef]
24. Pang, B.; Gu, S.; Zhang, Q.; Zhang, N.; Xiang, W. CCOS: A Coded Computation Offloading Strategy for Satellite-Terrestrial Integrated Networks. In Proceedings of the 2021 International Wireless Communications and Mobile Computing (IWCMC), Harbin, China, 28 June–2 July 2021; pp. 242–247. [CrossRef]
25. Wang, B.; Li, X.; Huang, D.; Xie, J. A Profit Maximization Strategy of MEC Resource Provider in the Satellite-Terrestrial Double Edge Computing System. In Proceedings of the 2021 IEEE 21st International Conference on Communication Technology (ICCT), Tianjin, China, 13–16 October 2021; pp. 906–912. [CrossRef]
26. 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Study on New Radio (NR) to Support Non-Terrestrial Networks (Release 15). 3GPP TR 38.811 V15.4.0. 2020. Available online: https://www.3gpp.org/ftp/Specs/archive/38_series/38.811. (accessed on 10 February 2022).
27. Lee, Y.; Choi, J.P. Connectivity Analysis of Mega-Constellation Satellite Networks With Optical Intersatellite Links. *IEEE Trans. Aerosp. Electron. Syst.* **2021**, *57*, 4213–4226. [CrossRef]
28. Tang, Q.; Fei, Z.; Li, B.; Han, Z. Computation Offloading in LEO Satellite Networks With Hybrid Cloud and Edge Computing. *IEEE Internet Things J.* **2021**, *8*, 9164–9176. [CrossRef]
29. Cui, G.; Li, X.; Xu, L.; Wang, W. Latency and Energy Optimization for MEC Enhanced SAT-IoT Networks. *IEEE Access* **2020**, *8*, 55915–55926. [CrossRef]
30. Zhao, Z.; Zhao, R.; Xia, J.; Lei, X.; Li, D.; Yuen, C.; Fan, L. A Novel Framework of Three-Hierarchical Offloading Optimization for MEC in Industrial IoT Networks. *IEEE Trans. Ind. Inform.* **2020**, *16*, 5424–5434. [CrossRef]
31. Ding, Y.; Li, K.; Liu, C.; Li, K. A Potential Game Theoretic Approach to Computation Offloading Strategy Optimization in End-Edge-Cloud Computing. *IEEE Trans. Parallel Distrib. Syst.* **2022**, *33*, 1503–1519. [CrossRef]
32. Monderer, D.; Shapley, L.S. Potential Games. *Games Econ. Behav.* **1996**, *14*, 124–143. [CrossRef]
33. Wang, E.; Dong, P.; Xu, Y.; Li, D.; Wang, L.; Yang, Y. Distributed Game-Theoretical Task Offloading for Mobile Edge Computing. In Proceedings of the 2021 IEEE 18th International Conference on Mobile Ad Hoc and Smart Systems (MASS), Denver, CO, USA, 4–7 October 2021; pp. 216–224. [CrossRef]
34. Fang, T.; Yuan, F.; Ao, L.; Chen, J. Joint Task Offloading, D2D Pairing, and Resource Allocation in Device-Enhanced MEC: A Potential Game Approach. *IEEE Internet Things J.* **2022**, *9*, 3226–3237. [CrossRef]