

# Cloud-edge collaborative computing offloading and resource allocation optimization in LEO satellite network

Yiwei Tian<sup>1, a</sup>, Xuebin Sun<sup>1, b \*</sup>

<sup>1</sup> Beijing University of Posts and Telecommunications, Key Laboratory of Universal Wireless Communication, Ministry of Education, Beijing, China.

<sup>a</sup> tianyiwei@bupt.edu.cn, <sup>b</sup> sunxuebin@bupt.edu.cn

**Abstract.** Satellite edge computing can provide communication and computing services to areas lacking ground network coverage, enhancing the computational capabilities of terrestrial users. However, due to the relatively limited computational resources available on satellites, cloud-edge collaborative computing in Low Earth Orbit (LEO) satellite communication scenarios emerges as a more optimal solution. In this paper, a cloud-edge collaborative offloading algorithm based on Deep Deterministic Policy Gradient (DDPG) algorithm is proposed to solve the offloading decision and resource allocation problems. The optimization problem is modeled as Markov decision process, and the comprehensive delay and energy consumption cost of computing tasks completed by ground users are optimized. Simulation results validate that the proposed algorithm has good convergence properties and lower comprehensive cost than processing all tasks locally and offloading all tasks to LEO satellites.

**Keywords:** LEO Satellite; Edge Computing; Computing Offloading; Resource Allocation.

## 1. Introduction

The deployment and application of the fifth generation of mobile communications (5G) have ushered in the era of the Internet of Things (IoT). To support network-wide coverage and high-speed mobility of users, and to expand the breadth and depth of current terrestrial network communication, the sixth generation (6G) is expected to encompass network infrastructures spanning space, air, and ground. It aims to integrate terrestrial and non-terrestrial networks to provide comprehensive coverage across air, space, and land domains[1]. Low Earth Orbit (LEO) satellites, compared to medium and high orbit satellites, offer advantages such as lower link loss, reduced latency, lower cost, and the capability of global coverage with multiple satellites[2]. With the continuous enhancement of satellite onboard capabilities, there has been a significant increase in the computational and storage capacities of satellites. Drawing inspiration from the concept of Mobile Edge Computing (MEC) in terrestrial networks, edge computing is being integrated into satellite networks[3][4][5]. In this approach, each satellite is treated as an edge node, enabling on-orbit edge computing. By processing data closer to the source, this method can effectively reduce the latency associated with satellite data transmission[6]. This development represents a pivotal shift in satellite network operations, moving towards more distributed and efficient data processing methods in space. Nowadays, the integration of MEC networks with LEO satellites has become a hot topic in both academia and industry[7], representing one of the key research directions for achieving future integrated space-air-ground systems[8].

At present, there have been some researches on the edge computing architecture and offload strategy in the low-orbit satellite communication scenario. Literature [9] designed a double-edge intelligent integrated satellite terrestrial network, where satellite MEC servers assist ground MEC servers in computing tasks offloaded by users. They proposed a task migration strategy based on a greedy algorithm to achieve load balancing and reduce system latency. Literature [10] proposes a hybrid cloud and edge computing LEO satellite (CECLS) network with a three-layer computing architecture. Under this network structure, ground users can choose to perform computing tasks locally or offload them to low-orbit satellites or cloud servers, and the optimization goal is to minimize energy consumption. A computational unloading algorithm using alternating direction multiplier method is proposed. In literature[11], a scheme based on potential game and Lagrange

multiplier method is proposed to obtain unloading decision and computing resource allocation under the satellite cloud-edge collaboration scenario. However, this paper takes the delay cost of ground users as the optimization objective, without considering the energy consumption of ground users.

## 2. System Model and Problem Formulation

### 2.1 Network model

As shown in Figure 1, the scenario considered in this paper is located in remote or disaster-affected areas without ground cellular network coverage. In this area,  $N$  ground user devices are randomly distributed, and each user device has a computationally intensive and non-divisible tasks. A LEO satellite equipped with a MEC server covers the entire area, providing communication and computing services to ground users in the region. There exists a communication link between the LEO satellite and the ground cloud server. Given the limited computational resources and energy on the satellite, users' computing tasks can be offloaded to the ground cloud servers for processing.

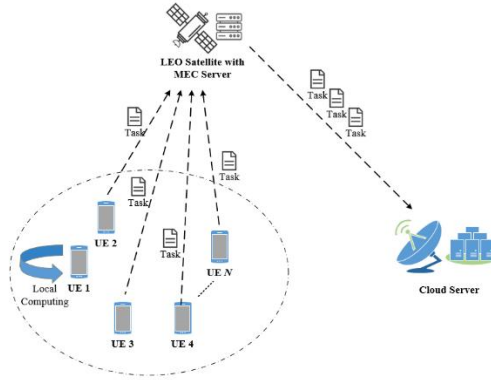


Figure 1 Network Structure Model

### 2.2 Communication Model

In this paper, it is assumed that the uplink between the ground user equipment and the LEO satellite employs Orthogonal Frequency Division Multiple Access (OFDMA) technology. Additionally, the communication band utilized is the Ka-band, and the free-space path loss model is considered. According to Shannon's law, the uplink data transmission rate from ground user  $n$  to the LEO satellite is:  $R_n^S = B_n^S \log_2 \left( 1 + \frac{p_n G_n^S L_S}{N_0} \right)$ , where  $B_n^S$  is the channel bandwidth allocated to ground user  $n$ ,  $p_n$  is the transmission power of ground user  $n$ ,  $G_n^S$  is the antenna gain between ground user  $n$  and the LEO satellite, and  $L_S$  is the free-space path loss[13]:  $L_S = \left( \frac{c}{4\pi f_s d_{n,s}} \right)^{d_e}$ , where  $c$  is the speed of light,  $f_s$  is the carrier frequency of the uplink space-to-ground link,  $d_{n,s}$  is the distance between user and satellite,  $d_e$  is the path loss exponent.

### 2.3 Computation Model

In this paper, the computational task generated by the ground user equipment  $n$  is denoted as  $C_n = \{D_n, \lambda_n\}$ , where  $D_n$  represents the size of the computational task generated and  $\lambda_n$  represents the computational resources required to complete 1 bit of the computational task  $n$  (cycles/bit).

The offloading decisions of the ground user equipment  $n$  are represented by  $x_{n,l} \in \{0,1\}$ ,  $x_{n,s} \in \{0,1\}$  and  $x_{n,c} \in \{0,1\}$ . When  $x_{n,l} = 1$ , it indicates that the task is entirely processed locally. When

$x_{n,s} = 1$ , it indicates that the task is entirely offloaded to the LEO satellite. When  $x_{n,c} = 1$ , it indicates that the task is entirely offloaded to the cloud data center for processing.

### 2.3.1 Local Computing

When the computing tasks are processed locally, the delay can be calculated by  $T_n^L = \frac{\lambda_n D_n}{f_n^L}$ , where  $f_n^L$  is the CPU computing frequency of user  $n$ . The energy consumption  $E_n^L$  can be calculated from the following formula[13]:  $E_n^L = \varepsilon(f_n^L)^2 \lambda_n D_n$ , where  $\varepsilon$  represents the energy factor, which depends on the chip architecture of the computing device.

### 2.3.2 LEO Satellite Computing

When users offload tasks to the LEO satellite, the tasks must first be transmitted to the satellite and then processed using the computational resources of the satellite's server. Since the data size of the computation results is very small, the transmission delay for returning the results to the ground user equipment can be considered negligible. Accordingly, the delay can be formulated as  $T_n^S = \frac{2d_{n,s}}{c} + \frac{D_n}{R_n^S} + \frac{\lambda D_n}{f_n^S}$ , where  $d_{n,s}$  is the distance between the user equipment  $n$  and the satellite,  $c$  is the speed of light,  $f_n^S$  is the computational resource allocated by the satellite to the user equipment  $n$ , and  $R_n^S$  is the uplink data transmission rate from ground user equipment  $n$  to the satellite.

The energy consumption  $E_n^S$  of user equipment  $n$  in this process, which arises from sending the computational task to the LEO satellite, can be expressed as:  $E_n^S = p_n \frac{D_n}{R_n^S}$ , where  $p_n^{\text{up}}$  is the transmission power of the ground user equipment.

### 2.3.3 Cloud Computing

When users offload tasks to be processed via LEO satellites, the tasks are first transmitted to the satellite, and then the satellite forwards them to the cloud server. Similar to the previous model, the transmission delay for returning the results to the user equipment is negligible. Therefore, the delay  $T_n^C$  incurred in this process includes: the transmission delay, the propagation delay, and the computation delay at the ground cloud server. The formula for the delay is:  $T_n^C = \frac{2d_{n,s}}{c} + \frac{D_n}{R_n^S} + \frac{2d_{s,c}}{c} + \frac{D_n}{R_n^C} + \frac{\lambda_n D_n}{f_n^C}$ , where,  $d_{s,c}$  is the distance between the LEO satellite and the satellite ground station,  $f_n^C$  is the computational resource allocated by the cloud server, and  $R_n^C$  is the data transmission rate from the LEO satellite to the satellite ground station.

The energy consumption of user equipment  $n$  in this process is  $E_n^C = p_n \frac{D_n}{R_n^S}$ , which is the energy consumption of sending the task to the LEO satellite.

## 2.4 Problem Formulation

The aim of this paper is to optimize the offloading decisions of ground user equipment, as well as the allocation of computational resources between satellites and cloud servers, to minimize the comprehensive cost of ground user equipment. This paper uses  $\mu \in [0, 1]$  as the weight factor for user delay and  $\rho = (1 - \mu)$  as the weight factor for user energy consumption, in order to balance the delay and energy consumption.

The objective function of the optimization problem in this paper can be formulated as:

$$\min_{\omega} \sum_{n=1}^N \mu(x_{n,l}T_n^L + x_{n,s}T_n^S + x_{n,c}T_n^C) + \rho(x_{n,l}E_n^L + x_{n,s}E_n^S + x_{n,c}E_n^C) \quad (1a)$$

$$\text{s.t.} \quad 0 \leq f_n^S \leq F_S, \forall n \in N \quad (1b)$$

$$0 \leq f_n^C \leq F_C, \forall n \in N \quad (1c)$$

$$\sum_{n=1}^N x_{n,s} f_n^S \leq F_S \quad (1d)$$

$$\sum_{n=1}^N x_{n,c} f_n^C \leq F_C \quad (1e)$$

$$x_{n,l}, x_{n,s}, x_{n,c} \in \{0,1\}, \forall n \in N \quad (1f)$$

$$x_{n,l} + x_{n,s} + x_{n,c} = 1, \forall n \in N \quad (1g)$$

The objective function (1a) aims to minimize this comprehensive cost. (1b), (1c), (1d), (1e): The computational resources allocated to each ground user from the satellite and cloud server must be within the available resources. (1f): These variables represent the binary offloading decisions for each ground user equipment  $n$ . (1g): Each ground user equipment  $n$  can only choose one location to complete its computational task.

### 3. DDPG-based Joint Optimization algorithm

The optimization problem in this paper involves both integer and continuous variables, making it challenging to obtain the optimal solution in polynomial time. Given the complexity of computational tasks, the scenario is modeled as a Markov Decision Process (MDP). A DDPG-based algorithm is proposed to optimize the comprehensive cost of tasks completed by user devices. The definition of state space, action space and reward function of Markov decision process in this scenario is given below.

**State Space:** The state space consists of each user device's transmission power  $p_n$  and computational tasks  $C_n = \{D_n, \lambda_n\}$ , represented as a one-dimensional vector:  $S_t = (p_1, p_2, \dots, p_n, D_1, D_2, \dots, D_n, \lambda_1, \lambda_2, \dots, \lambda_n)$ .

**Action Space:** The action space includes each user's offloading decision, computational resources allocated by the LEO satellite, and resources allocated by the ground cloud server, represented as a vector:  $A_t = (x_1, x_2, \dots, x_n, f_1^S, f_2^S, \dots, f_n^S, f_1^C, f_2^C, \dots, f_n^C)$ .

**Reward Function:** Assuming  $C_{\text{local}}$  as the comprehensive cost of local task processing and  $C_{\text{off}}$  as the cost corresponding to the offloading action, the reward function is defined as  $R_t = (C_{\text{local}} - C_{\text{off}})/C_{\text{local}} + p$ , where  $p$  is a penalty for not meeting the constraints outlined in the previous chapter.

The proposed DDPG-based optimization algorithm consists of four networks: Actor, Actor Target, Critic, and Critic Target networks. In this approach, the agent initially receives state information from the environment. After obtaining the initial state, it inputs this into the Actor network to get the action  $A_t$  and calculate the reward, leading to the next state  $S_{t+1}$ . The tuple  $\{S_t, A_t, R_t, S_{t+1}\}$  is stored in an experience replay pool. During training, a batch of experiences is randomly sampled from the pool to update the Critic network, aiming to align its predicted Q-value with the actual return. The Critic network's assessment guides the Actor network's update, enabling the selection of actions that yield higher Q-values. Target networks are used to stabilize the training process through slow updates of their weights, smoothing the learning objectives. Through this method, the algorithm effectively learns policies in the action space, optimizing offloading strategies and computational resource allocation decisions.

### 4. Simulation Results

To validate the performance of the proposed algorithm, this paper constructs a simulation environment using Python and develops the reinforcement learning algorithm model using TensorFlow.

## 4.1 Parameter setting

In the simulation of this paper, considering a nearby area with 10 ground user devices, the orbital height of the LEO satellite is set at 780 km. The elevation angle of user devices is 45 degrees. The transmission power of user devices is randomly distributed between 0.6W and 0.8W. The size of computational tasks generated by user devices is randomly distributed between 8 Mbit and 12 Mbit, with a computational intensity of 300 CPU cycles/bit. The local computing CPU frequency of user devices is 0.5 Gcycles/s. The computational resources of the LEO satellite's onboard server are 10 Gcycles/s, and the ground cloud server's computational resources are 20 Gcycles/s. The bandwidths of both the uplink and downlink are 800 MHz, with an uplink carrier frequency of 30 GHz and a downlink carrier frequency of 20 GHz. The weight factor  $\mu$  is 0.5.

## 4.2 Simulation Results

### 4.2.1 Convergence analysis

Figure 2 presents the reward value variation curve of the proposed algorithm with the number of training episodes. From the figure, we can observe that in the initial few episodes, the reward values are relatively low, indicating that the agent is exploring the environment and learning basic strategies. After the initial phase, the reward values rise rapidly, suggesting that the agent has learned effective strategies to obtain higher rewards. Following this rapid increase, the reward values stabilize and show minimal fluctuations, implying that the agent's strategy is nearing optimality and is relatively stable. Around 100 episodes later, the reward values maintain a high level with hardly any significant fluctuations, typically indicating that the learning has converged. Therefore, the proposed optimization algorithm based on DDPG demonstrates good convergence properties.

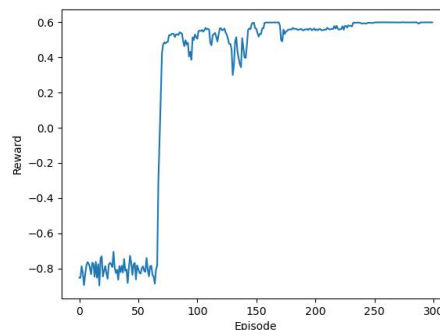


Figure 2 Convergence performance

### 4.2.2 Performance Comparisons

In this paper, the proposed algorithm is compared with several other strategies: processing all tasks locally (LC), offloading all tasks to LEO satellites (LEOC), and randomly offloading tasks to either Low Earth Orbit satellites or cloud servers (RANDOM).

Figure 3 illustrates the variation in total cost for different offloading strategies as the number of users increases. With the rise in user numbers, the comprehensive cost for users increases under all strategies. Due to the LC algorithm being unaffected by offloading decisions and resource allocation outcomes, its system-wide comprehensive cost exhibits a linear growth with the increase in users. Moreover, as the number of users increases, the performance advantage of the proposed strategy over the LEOC algorithm becomes increasingly evident. This is because, with more users, the LEOC algorithm is limited by the satellites' finite computational and communication resources. The increase in users leads to a decrease in the data transmission rate to Low Earth Orbit satellites and a reduction in the computational resources allocated to each task on the satellites, resulting in a significant increase in the delay for the LEOC algorithm. In contrast, the proposed algorithm efficiently allocates onboard computational resources and balances the number of tasks offloaded to

satellites and ground-based cloud servers. It maximizes the utilization of the entire system's communication and computational resources, thereby demonstrating superior performance.

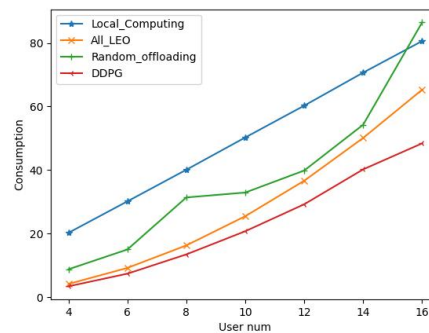


Figure 3 Total consumption with different number of users

## 5. Conclusion

In the context of LEO satellite communication networks, an algorithm for cloud-edge collaborative computing offloading and computational resource allocation has been proposed based on DDPG. This algorithm takes into account the differences in computational tasks and transmission power among various users, aiming to minimize the latency and energy cost of users completing computational tasks. The simulation results demonstrate that, compared to benchmark algorithms, the proposed method effectively reduces the overall latency and energy consumption for ground users. For future research, it would be beneficial to consider optimization of offloading algorithms for partial task offloading in order to further reduce the overall consumption in such scenarios.

## References

- [1] ZHANG Ping, XU Wenjun, WANG Fengyu, et al. Intellicise Air-Space-Ground Integrated Networks[J]. Radio Communications Technology, 2022, 48(3):381-384.
- [2] ZHANG Youzhi. Construction and Conceiving of LEO Communication Satellite Constellation[J]. Computer & Network, 2020.
- [3] Zhang S , Cui G , Long Y ,et al. Joint Computing and Communication Resource Allocation for Satellite Communication Networks with Edge Computing[J].China Communications, 2021.
- [4] Z. Song, Y. Hao, Y. Liu, and X. Sun, Energy-efficient multiaccess edge computing for terrestrial-satellite Internet of Things[J]. IEEE Internet Things J., vol. 8, no. 18, pp. 14202–14218, Sep. 2021.
- [5] X. Gao, R. Liu, A. Kaushik, and H. Zhang, Dynamic resource allocation for virtual network function placement in satellite edge clouds[J]. IEEE Trans. Netw. Sci. Eng., vol. 9, no. 4, , Jul./Aug. 2022.
- [6] Denby Bradley, Lucia Brandon. Orbital edge computing: machine inference in space[J].IEEE Computer Architecture Letters,2019,18(1):59-62.
- [7] TANG Qinqin, XIE Renchao, LIU Xu,et al. MEC enabled satellite-terrestrial network: architecture, key technique and challenge [J]. Journal on Communications, 2020, 41(4):20.
- [8] FENG Jianyuan, LI Jie, PANG Lixin, et al. Architecture design of satellite networks in integrated terrestrial-satellite network[J].Telecommunication Engineering, 2019, 59(7):775-779.
- [9] Zhang J , Zhang X , Wang P ,et al. Double-Edge Intelligent Integrated Satellite Terrestrial Networks[J]. China Communications, 2020, 17(9):19.DOI:CNKI:SUN:ZGTO.0.2020-09-014.
- [10] Q. Tang, Z. Fei, B. Li and Z. Han, Computation Offloading in LEO Satellite Networks With Hybrid Cloud and Edge Computing[J]. IEEE Internet of Things Journal, vol. 8, no. 11, pp. 9164-9176, 1 June1, 2021

- [11] Tong M , Wang X , Li S ,et al. Joint Offloading Decision and Resource Allocation in Mobile Edge Computing-Enabled Satellite-Terrestrial Network[J].Symmetry, 2022, 14.DOI:10.3390/sym14030564.
- [12] Ebert B. R. Introduction to satellite communication[M]. Artech House, 2008.
- [13] Miettinen A P, Nurminen J K. Energy efficiency of mobile clients in cloud computing[J]. USENIX Association, 2010. p. 4.