**Proceedings of the Estonian Academy of Sciences**
2026, **75**, 1, 56–74

https://doi.org/10.3176/proc.2026.1.06

www.eap.ee/proceedings
Estonian Academy Publishers

SPACE-AIR-GROUND NETWORK, WIRELESS COMMUNICATION, COMPUTING OFFLOADING

RESEARCH ARTICLE

Received 19 September 2025
Accepted 10 December 2025
Available online 20 February 2026

**Keywords:**
Internet of Things, wireless communication, space-air-ground integrated network, computing offloading, resource allocation

**Corresponding author:**
Guoqiang Zheng
lyzhenggq@126.com

**Citation:**
Li, Z., Zheng, G., Han, Z., Xia, P., Ma, H. and Ji, B. 2026. Joint task offloading and resource allocation for remote IoT in space-air-ground networks architecture. *Proceedings of the Estonian Academy of Sciences*, **75**(1), 56–74. https://doi.org/10.3176/proc.2026.1.06



# Joint task offloading and resource allocation for remote IoT in space-air-ground networks architecture

Zhenhua Li, Guoqiang Zheng, Zhe Han, Pingjie Xia, Huahong Ma and Baofeng Ji

College of Information Engineering, Henan University of Science and Technology, Luoyang 471023, China

## ABSTRACT

In remote areas with insufficient ground infrastructure, user devices (UDs) are constrained by limited computing resources, which poses substantial challenges to achieving low-latency and energy-efficient data processing. To address these issues, this paper proposes a dual-layer heterogeneous network architecture that makes full use of unmanned aerial vehicle (UAV) and low Earth orbit (LEO) computing resources. Considering the high mobility of LEO satellites, the characteristics of channel variations, and the queuing delay of user task offloading, the optimization objective is modeled as a mixed-integer nonlinear programming problem, aiming to minimize the weighted sum of delay and energy consumption (i.e., the total system cost). A low-complexity alternating optimization algorithm is proposed. The original problem is decomposed into three subproblems: bandwidth allocation, central processing unit (CPU) frequency allocation, and task scheduling optimization, which are solved using convex optimization, the Lagrange multiplier method, and the alternating direction method of multipliers (ADMM), respectively. Finally, the Pareto method is used to seek the best trade-off among the optimization objectives. The simulation results indicate that the average total system cost of the alternating optimization for task offloading and resource allocation (AOTORA) decreases by 25.8%, 11.63%, 12.48%, and 6.84% compared with random optimization, equal bandwidth allocation, the offloading LEO satellite algorithm, and distributionally robust optimization, respectively.

## 1. Introduction

With the rapid advancement of mobile communication and network technology, user devices (UDs) face growing demands for low latency, high energy efficiency, and enhanced experience, requiring sufficient computational capacity for efficient task processing [31]. New standards and technologies for next-generation mobile systems that include Internet of Things (IoT), cloud computing, and edge computing [13,14,25,29] are being proposed and implemented, while cloud computing is highly dependent on the network, and centralized management tends to lead to competition for users' resources, which in turn triggers service disruptions or higher response latency. The Cloudlet concept [30] proposes the concept of first edge computing, which means offloading computation and storage to the periphery of the network (e.g., edge nodes) adjacent to UDs in order to provide computation resources and storage to nearby UDs [23].

However, due to the limitation of network capacity and coverage, the traditional terrestrial edge computing system cannot meet the computing needs of Internet of Remote Things (IoRT) users in remote areas such as oceans and mountains [19]. Fortunately, space-air-ground integrated networks (SAGIN) constitute a vital part of future communication systems, offering seamless coverage and adaptability to diverse services. SAGIN utilizes a heterogeneous architecture and consists of three network segments: satellite, airborne, and ground segments [6]. For the airborne part, unmanned aerial vehicles (UAVs) serve as an effective auxiliary device, offering flexible deployment, broad coverage, and low cost [38]. For the satellite part, the low Earth orbit

(LEO) satellite networks provide extensive coverage and extend connectivity to remote regions such as oceans and mountains, enabling reliable long-term communications for ground users. In addition, the network architecture exhibits significant complexity. The link quality and coverage of aerial networks vary dynamically, resulting in fluctuations in server availability and communication latency. Moreover, distinct network segments within SAGIN possess heterogeneous link qualities and capacity constraints, making the design of efficient computational offloading strategies a critical challenge [8,28].

In the study of ground- or UAV-assisted computation offloading and resource allocation, Li et al. [16] aimed to minimize the overall system cost and proposed a method that combines partial offloading with collaborative mobile edge computing (MEC). They transformed the problem into a multi-objective optimization problem and solved it using a two-tier alternating optimization framework. Su et al. [34] proposed a collaborative optimization framework that jointly considers task offloading and UAV trajectory design to minimize the energy consumption of UDs. Shen et al. [32] presented a novel UAV-based computation offloading scheme aimed at minimizing both the mean task completion latency and energy consumption in vehicular networks. Qi et al. [27] introduced a dual collaborative computation offloading strategy leveraging multiple UAVs in a MEC framework, aiming to optimize user scheduling of UAV paths to improve the system's energy efficiency. Zhong et al. [44] proposed a dual UAV-supported edge computing system aimed at minimizing the total user latency by optimizing UAV trajectories, computational resource allocation, and UAV function switching. Although Li et al. [16] optimized the computing costs for ground terminal devices, the scalability is poor, and the deployment locations of edge servers are fixed with a limited coverage radius. In addition, research work in [16,27,32,44] mainly focuses on single-layer network architectures, which have limitations in both space and resources: first, it ignores the collaborative architecture between UAVs and LEO satellites; second, drones are constrained by hardware resources, and their limited onboard computing capabilities make efficient trajectory planning in complex environments such as oceans and mountains challenging.

Therefore, we explore the distribution of computational tasks and resource management in the integrated dual-layer system of UAVs and LEO satellites. Chen et al. [4] introduced a space-centric approach for offloading computational tasks, considering the limited coverage time and limited resources of LEO satellites for maintaining service continuity, minimizing the overall execution cost by optimizing the offloading decision and power control. Li et al. [15] investigated low-energy UAVs and LEO satellites-assisted mobile edge computing in SAGIN, aiming to optimize the long-term energy performance of UAV machines and LEO satellites by jointly optimizing task scheduling, UAV trajectories, and bandwidth allocation. Zhang et al. [22] and Liu et al. [41] studied the task offloading mechanism under the SAGIN architecture, but with different focuses. The former considered the dynamics of the device in a comprehensive manner, while the latter focused on power-constrained IoT device scenarios. Li et al. [17] proposed a new space-air-ground-sea multi-layer computing offloading architecture, using high-altitude platform stations as relay nodes and optimizing task offloading and resource allocation through multi-agent proximal policy optimization (MAPPO) to minimize multi-objective issues such as latency, energy consumption, and operational costs. Most of the existing studies employ intelligent algorithms such as deep reinforcement learning (DRL), multi-agent stochastic learning (MASL), or MAPPO to address the optimization problem. However, the scenarios considered in this work involve complex environments, such as mountainous regions. Since DRL relies on deep neural networks, it is difficult to deploy on resource-constrained platforms such as UAVs. Moreover, training such models may take from several days to months, which fails to meet the real-time requirements of UDs.

In comparison, alternating optimization, as an efficient low-complexity algorithm, provides a more concrete and feasible solution approach for resource-constrained dynamic environments. He et al. [10] explored the data offloading problem in SAGIN and proposed an approximation strategy to optimize task scheduling and power control, balancing energy consumption and completion time. Mao et al. [24] presented a joint space-air cloud-edge computing structure aiming to minimize the maximum computational latency between IoT devices through the joint optimization of correlation control, transmission power, task allocation, and bandwidth distribution. Sriharsha et al. [33] proposed a combination of terrestrial fixed MEC and UAV relaying to minimize the total energy consumption of the user equipment by optimizing UAV trajectory and connection scheduling. Mei et al. [26] proposed a space-air-ground cooperative network using non-orthogonal multiple

access (NOMA) technology, employing a block coordinate descent method to decompose the original problem into multiple interrelated coupled subproblems, jointly optimizing power control and computation frequency allocation, as well as task offloading strategies, aiming to minimize system energy consumption. Existing works optimize the energy consumption or latency of UDs, but ignore the limited computing power or coverage time of LEO satellites [10,24,26,33].

In addition, in complex dynamic scenarios, Xie et al. [39] proposed an enhanced snow ablation optimization algorithm (ESAO), which significantly enhances convergence speed and global search capability by introducing an adaptive T-distribution, Cauchy mutation, and leader boundary control strategy. Liu et al. [20] demonstrated how to combine hybrid neural network models with metaheuristic optimization strategies to effectively address nonlinear system prediction tasks. Liu et al. [21] further emphasized the importance of implementing customized evolutionary optimization in complex energy management systems. ESAO in Xie et al. [39] and alternating optimization belong to different optimization paradigms, but ESAO needs to gradually converge to a satisfactory solution through hundreds or even thousands of iterations, at the cost of higher computational complexity. This study focuses on the optimized allocation of static resources, providing a fundamental solution for resource management in space-air-ground integrated networks. However, real-world network environments are highly dynamic, and the next challenge this research faces is inherently related to the existing outstanding work. First, in terms of temporal dynamics, the task of randomly arriving problems is similar to the oil temperature prediction problem addressed by Liu et al. [20], with the core issue being how to accurately characterize a dynamically changing process. Second, in terms of the nature of the problem, this study, similarly to another work by Liu et al. [21], falls under the category of multi-objective optimization under constrained resources, laying a foundation for later leveraging of their advanced hybrid optimization algorithm frameworks (such as DE-HHO) to address more complex dynamic scheduling scenarios in this problem.

Based on the above research, to address communication and computing needs in remote areas and emergency scenarios, this paper constructs an space-air-ground network in which a drone swarm carrying edge servers collaborates with LEO satellites. In this network architecture, drones provide temporary and rapid computing services, while satellites ensure broader communication coverage. However, the high mobility of satellites results in limited coverage time and periodically varying channels, posing challenges for heterogeneous network resource management. Therefore, we propose a joint optimization problem to minimize the total computation cost for ground users (weighted sum of computation delay and energy consumption) by jointly optimizing bandwidth allocation, computing resource allocation, and task offloading scheduling, under the constraints of maximum workload and communication resources. The main contributions are summarized as follows:

1. A dual-layer heterogeneous network system that fully utilizes UAV and LEO computing resources is proposed. Considering the high mobility of LEO, channel variation characteristics, and user task offloading queuing delay, the optimization objective is modeled as a mixed-integer nonlinear programming problem, aiming to minimize the weighted sum of latency and energy consumption.

2. To address this issue, we propose a low-complexity alternating optimization algorithm. The original problem is decomposed into three subproblems: bandwidth allocation, central processing unit (CPU) frequency allocation, and task scheduling optimization. Convex optimization, the Lagrange multiplier method, and the alternating direction method of multipliers (ADMM) are used to solve them, respectively, and finally, the Pareto approach is applied to seek the best trade-off among the optimization objectives.

3. Simulation results demonstrate the convergence and superiority of the proposed alternating optimization for task offloading and resource allocation (AOTORA) algorithm. Compared with other benchmark schemes (random optimization, equal bandwidth allocation, full satellite offloading, and distributed robust optimization), the average total computational cost per user is reduced by 25.8%, 11.63%, 12.48%, and 6.48%, respectively.

The remainder of this paper is organized as follows: Section 2 presents the system models, Section 3 presents the formulation and analysis of the problem, Section 4 develops the AOTORA algorithm to solve the problem, Section 5 provides an analysis of the simulation results, Section 6 summarizes the research findings, discusses limitations, and outlines future research directions.

## 2. System model and problem description

### 2.1. Network model

This paper considers a dual-layer space-air-ground internet of the IoRT network, and the specific system model is illustrated in Fig. 1a. The space-based network is composed of $L$ LEO satellites equipped with edge servers, one of which is selected to establish a communication link with the ground, and the set of LEO satellites is denoted by $l \in \mathcal{L} = \{1, 2, \ldots, l, \ldots, L\}$. LEO satellite communication links are updated periodically over time. The air-based network consists of $N$ UAVs equipped with edge servers, which are fixed in position and hover in the air, and the set is represented by $n \in \mathcal{N} = \{1, 2, \ldots, n, \ldots, N\}$. Both satellites and drones can provide computing services to ground-based devices. The terrestrial network consists of $M$ UDs, and the set is represented by $m \in \mathcal{M} = \{1, 2, \ldots, m, \ldots, M\}$.

Depending on the UDs, the computational tasks take different computational approaches, which can be offloaded to the LEO satellites or forwarded to the UAVs for processing by the ground users via wireless links. Assuming that each UD has only one computational task $J_m = \{D_m, \lambda_m\}$, where $D_m$ denotes the task data size of the UD and $\lambda_m$ denotes the CPU frequency cycle needed to process a single bit of data. Since tasks cannot be split, UDs must choose between UAVs and LEO satellites as offloading targets [36]. In addition, considering the variable channel environment among the UDs and the LEO satellites, UDs need to select the appropriate LEO satellites for mission computation. Table 1 provides an overview of the main symbols of the system.

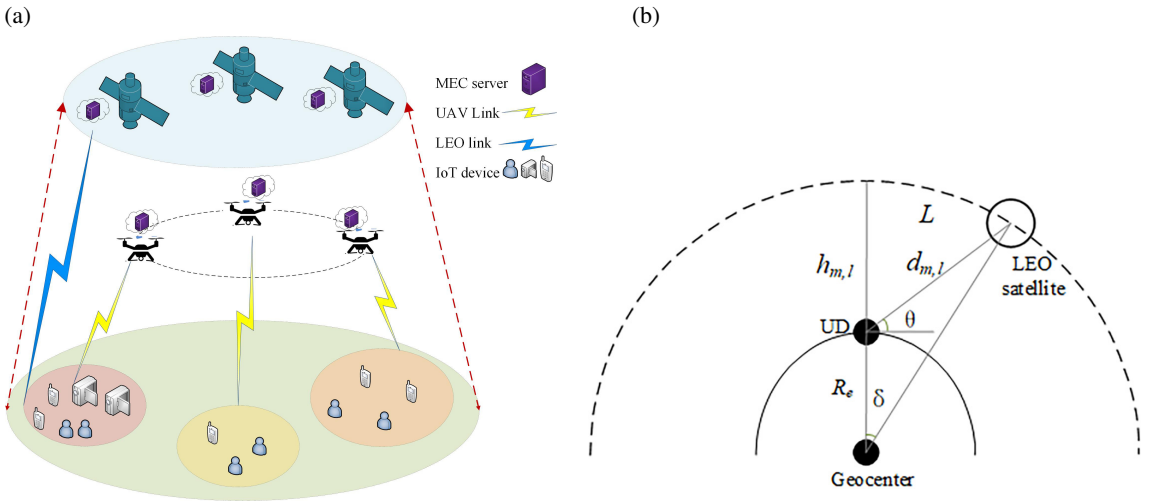


**Fig. 1.** System network model: UD network model (a) and the position relationship between UD and LEO satellite (b).

### 2.2. Coverage model

The high maneuverability of LEO satellites results in dynamic variations in their coverage areas, which makes it challenging to maintain continuous and stable communication links. Therefore, the satellite coverage model must account for these dynamic changes. According to [36], communication and computation can only occur when specific geometric relationships are met (see Fig. 1b). LEO satellites operate in predetermined orbits, $h_{m,l}$ represents the distance from the UDs to the LEO satellite orbit, $R_e$ denotes the Earth's radius, and $d_{m,l}$ is the distance between the UDs and the LEO satellite. $\theta$ represents the elevation angle of the UDs, and the solving formula is

$$\theta = \arccos\left(\frac{R_e + h_{m,l}}{d_{m,l}} \cdot \sin\delta\right). \tag{1}$$

The geocentric angle $\delta$ from the Earth's center covering the LEO satellite's range is calculated as

$$\delta = \arccos\left(\frac{R_e}{R_e + h_{m,l}} \cdot \cos\theta\right) - \theta. \tag{2}$$

**Table 1.** System variables

| Notation | Definition |
|---|---|
| $M/N/L$ | Number of UDs/UAVs/LEO satellite |
| $J_m = \{D_m, \lambda_m\}$ | UD $m$ task data size and CPU required for computation |
| $d_{m,l}$ | Distance between UD $m$ and LEO satellite $l$ |
| $h_{m,l}$ | Distance between UD $m$ and satellite orbit |
| $H_m^n / H_m^l$ | Channel gains from UD $m$ to UAV $n$ and to satellite $l$ |
| $R_{m,n} / R_{m,l}$ | Distance between UD $m$ and satellite orbit |
| $B^n / B^l$ | Bandwidth of UAV $n$ and satellite $l$ |
| $\beta_{m,n}$ | Bandwidth coefficient of UAV $n$ allocated to UD $m$ |
| $f_{m,n} / f_{m,l}$ | Computational resources from UAV $n$ and satellite $l$ to UD $m$ |
| $F_n^{\max} / F_l^{\max}$ | Maximum computational resources of UAV $n$ and satellite $l$ |
| $p_{m,n} / p_{m,l}$ | Uplink transmit power of UD $m$ |
| $p^n / p^l$ | Launch power of UAV $n$ and satellite $l$ |
| $t_{m,n}^{\text{up}} / t_{m,n}^{\text{com}} / t_{m,n}^{\text{wait}}$ | Transmission, computation, and wait time between task $D_m$ and UAV $n$ |
| $E_{m,n}^{\text{up}} / E_{m,n}^{\text{com}}$ | Transmission and computation energy consumption between task $J_m$ and UAV $n$ |
| $t_{m,l}^{\text{up}} / t_{m,l}^{\text{com}} / t_{m,l}^{\text{delay}} / t_{m,l}^{\text{wait}}$ | Transmission, computation, propagation, and wait time between task $J_m$ and satellite $l$ |
| $E_{m,l}^{\text{up}} / E_{m,l}^{\text{com}}$ | Transmission and computation energy consumption between task $J_m$ and satellite $l$ |
| $t_{m,n} / t_{m,l}$ | Computation time of task $J_m$ on UAV $n$ or satellite $l$ |
| $E_{m,n} / E_{m,l}$ | Computation energy consumption of task $J_m$ on UAV $n$ or satellite $l$ |
| $a_{m,n} / a_{m,l}$ | Offloading decisions from UD $m$ to UAV $n$ and satellite $l$ |
| $T_m / E_m$ | Total computation delay and total energy consumption of all tasks |
| $\gamma$ | Lagrange multiplier vector |
| $\rho$ | Penalty parameter |
| $\bar{\omega}$ | Weight factor |
| $\sigma^2$ | Noise power |
| $T_{\max} / T_{\min}$ | Maximum and minimum total delay required for all tasks |
| $E_{\max} / E_{\min}$ | Maximum and minimum total energy consumption for all tasks |

Therefore, the maximum communication time between the UD and LEO satellite is

$$T_c = \frac{L}{v_s}, \tag{3}$$

where $v_s$ represents the satellite's velocity, and $L$ represents the arc length of the communication time between the UD and LEO satellite, which can be calculated by

$$L = 2 \cdot \left(R_e + h_{m,l}\right) \cdot \delta. \tag{4}$$

The satellite orbit altitude $h_{m,l}$ and the Earth's radius $R_e$ are system constants. The orbital velocity of a satellite is $v_s$, determined by the orbital altitude, and the satellite is in the same orbit, which is also a constant for near-circular orbits. Most importantly, the geocentric angle $\delta$ proved to be a self-consistent and definite mathematical solution through Eq. (2). Once the satellite and user position are determined, the coverage angle $\delta$ is a fixed value, resulting in the communication arc length $L$ and the final time $T$ also being fixed. An update cycle is set, and after each cycle, the system will re-compare the channel conditions of each satellite and determine the optimum. We choose satellite coverage time constraints, periodically updated channels, and fixed-interval mission data volume randomness as modeling starting points, which aim to focus on solving the core complexity of NP-hard resource allocation.

### 2.3. Communication model

Given that the size of computational results is significantly smaller than that of input data, the downlink transmission rate is much higher than the uplink rate. This paper focuses solely on uplink transmission latency and energy consumption, while neglecting the downlink latency and energy overhead from satellites and UAVs to UDs. Both the ground devices' and the UAVs' positions are fixed, the location of UD $m$ is represented as $W_m = (x_m, y_m, 0)$, the UAV hovers at an altitude of $h$ with the coordinates $W_n = (x_n, y_n, h)$, and the channel gain between UD $m$ and UAV $n$ is denoted as

$$H_m^n = \frac{\gamma_0}{\left[\sqrt{(x_m - x_n)^2 + (y_m - y_n)^2 + h^2}\right]^\kappa}, \tag{5}$$

where $\gamma_0$ is the channel power gain at a reference distance of 1 m and $\kappa$ is the path loss exponent for the UD–UAV link.

When a ground-based IoT UD selects a LEO satellite for task offloading, the satellite begins to perform in-orbit computations for the upcoming task. All LEO satellites operate at the same orbital altitude $h_{m,l}$, and the UD $m$ and LEO satellite $l$ channels are modeled by

$$H_m^l = \frac{G\lambda}{4\pi d_{m,l}(t)} \cdot h_{\text{shadow}} \cdot h_{\text{multipath}}, \tag{6}$$

where $G$ indicates the antenna gain, $\lambda$ denotes the signal wavelength, $d_{m,l}(t)$ denotes the distance between UD $m$ and LEO satellite $l$ in the same cycle, and $h_{\text{shadow}}$ and $h_{\text{multipath}}$ denote the shadow fading and multipath effects, respectively.

UD $m$ transmits upward through the wireless channel. When UD $m$ connects to UAV $n$ or LEO satellite $l$, the uplink transmission rates are respectively expressed as

$$R_{m,n} = \beta_{m,n} B^n \log_2\left(1 + \frac{H_m^n p_{m,n}}{\sum\limits_{i \neq m} H_i^n p_{i,n} + \sigma^2}\right), \tag{7}$$

$$R_{m,l} = B^l \log_2\left(1 + \frac{p_{m,l} H_m^l}{\sigma^2}\right), \tag{8}$$

where $\beta_{m,n}$ represents the bandwidth allocation coefficient assigned by UAV $n$ to UD $m$, and $B^m$, $B^l$ denote the available channel bandwidths for UAV $n$ and LEO satellite $l$, respectively. The terms $p_{m,n}$, $p_{m,l}$ correspond to the transmission powers between UD $m$ and UAV $n$, as well as between UD $m$ and LEO satellite $l$. $\sum\limits_{i \neq m} H_i^n p_{i,n}$ denotes the total interference caused by other terrestrial UDs transmitting data from UD $m$ to UAV $n$ based on the same channel, and $\sigma^2$ signifies the background noise inherent in communications between ground UDs and other devices.

### 2.4. Computation model

For UDs, two computational offloading schemes are available. The binary variables $a_{m,n}$ and $b_{m,l}$ are defined as the offloading decision variables for ground UDs, where $a_{m,n} = 1$ signifies that UD $m$ is allocated to UAV $n$, otherwise $a_{m,n} = 0$. Similarly, $b_{m,l} = 1$ indicates that UD $m$ is allocated to satellite $l$, otherwise $b_{m,l} = 0$. Determine the optimal UDs' offloading selection by minimizing the UDs' total computational cost.

(1) UAV computing: Within the UAV computing framework, the computational task $J_m$ is transferred to UAV $n$ for execution. The computational power (CPU cycles/s) allocated by UAV $n$ to UD $m$ is denoted as $f_{m,n}$, which may vary across different UD $m$. The maximum computational power of UAV $n$ is denoted as $F_n^{\max}$. The processing latency of the computational task $J_m$ by UAV is denoted as

$$t_{m,n}^{\text{up}} = \frac{D_m}{R_{m,n}}, \tag{9}$$

$$t_{m,n}^{\text{com}} = \frac{\lambda_m}{f_{m,n}}, \tag{10}$$

where $t_{m,n}^{\text{up}}$ and $t_{m,n}^{\text{com}}$ represent the transmission latency and computational latency of user task $J_m$ to UAV $n$, respectively.

Additionally, if the UAV resources are currently occupied, the waiting delay for the next unloading task is $t_{m,n}^{\text{wait}}$. The total execution time for task $J_m$ from UD $m$ to UAV $n$ is

$$t_{m,n}^{\text{com}} = t_{m,n}^{\text{up}} + t_{m,n}^{\text{com}} + t_{m,n}^{\text{wait}}. \tag{11}$$

The energy consumption of task $J_m$ computed from UAV is denoted as

$$E_{m,n}^{\text{up}} = p^m t_{m,n}^{\text{up}}, \tag{12}$$

$$E_{m,n}^{\text{com}} = p^n \cdot \frac{\lambda_m}{f_{m,n}}, \tag{13}$$

$$E_{m,n} = E_{m,n}^{\text{up}} + E_{m,n}^{\text{com}}, \tag{14}$$

where $p^m$ and $p^n$ are the transmit powers of UD $m$ and UAV $n$. $E_{m,n}^{\text{up}}$, $E_{m,n}^{\text{com}}$, and $E_{m,n}$ denote the corresponding transmission energy consumption, computational energy, and total energy consumption of task $J_m$ on UAV $n$.

(2) LEO satellite computing: Within the satellite computing framework, the computational task $J_m$ is transferred to satellite $l$ for execution. The computational power (CPU cycles/s) allocated by satellite $l$ to UD $m$ is denoted as $f_{m,l}$, which can be different for different UD $m$. The maximum computational power of satellite $l$ is denoted as $F_l^{\max}$. However, the communication process between UDs and the satellite is subject to propagation latency. The execution latency of the computational task $J_m$ by satellite $l$ is denoted as

$$t_{m,l}^{\text{up}} = \frac{D_m}{R_{m,l}}, \tag{15}$$

$$t_{m,l}^{\text{delay}} = \frac{d_{m,l}}{c}, \tag{16}$$

$$t_{m,l}^{\text{com}} = \frac{\lambda_m}{f_{m,l}}, \tag{17}$$

where $d_{m,l}$ is the distance between UD $m$ and satellite $l$, $c$ is the speed of light, and $t_{m,l}^{\text{up}}$, $t_{m,l}^{\text{delay}}$, and $t_{m,l}^{\text{com}}$ represent the transmission latency, propagation latency, and computational latency, respectively.

Additionally, if the LEO satellite resources are currently occupied, the waiting delay for the next unloading task is $t_{m,l}^{\text{wait}}$. The total execution time for task $J_m$ from UD $m$ to LEO $l$ is

$$t_{m,l} = t_{m,l}^{\text{up}} + t_{m,l}^{\text{delay}} + t_{m,l}^{\text{com}} + t_{m,l}^{\text{wait}}. \tag{18}$$

The energy consumption of task $J_m$ computed from satellite $l$ is denoted as

$$E_{m,l}^{\text{up}} = p^m t_{m,l}^{\text{up}}, \tag{19}$$

$$E_{m,l}^{\text{com}} = p^l \cdot \frac{\lambda_m}{f_{m,l}}, \tag{20}$$

$$E_{m,l} = E_{m,l}^{\text{up}} + E_{m,l}^{\text{com}}, \tag{21}$$

where $p^l$ is the transmit power of LEO satellite $l$. $E_{m,l}^{\text{up}}$, $E_{m,l}^{\text{com}}$, and $E_{m,l}$ denote the corresponding transmission energy consumption, computational energy consumption, and total energy consumption of task $J_m$ on satellite $l$.

## 3. Problem description

Based on the computational models above, the total delay and total energy consumption required for system computing tasks are expressed as follows:

$$T_m = \sum_{m \in \mathcal{M}} a_{m,n} t_{m,n} + b_{m,l} t_{m,l}, \tag{22}$$

$$E_m = \sum_{m \in \mathcal{M}} a_{m,n} E_{m,n} + b_{m,l} E_{m,l}. \tag{23}$$

We studied the joint optimization problem of bandwidth allocation, computational resource allocation, and user scheduling, aiming to minimize total system cost, which is formulated as

$$\begin{aligned}
\text{P: } & \min_{a,b,\beta,f} \sum_{m\in\mathcal{M}} \{\bar{\omega}T_m + (1-\bar{\omega})E_m\} \\
& \text{s.t. C1: } \{a_{m,n}, b_{m,l}\} \in \{0,1\}, \forall m, n, \\
& \quad \text{C2: } \sum_{n\in\mathcal{N}} a_{m,n} + \sum_{l\in\mathcal{L}} b_{m,l} = 1, \forall m, \\
& \quad \text{C3: } 0 \le f_{m,n} \le F_n^{\max}, \sum_{m\in\mathcal{M}} f_{m,n} \le F_n^{\max}, \\
& \quad \text{C4: } 0 \le f_{m,l} \le F_l^{\max}, \sum_{m\in\mathcal{M}} f_{m,l} \le F_l^{\max}, \\
& \quad \text{C5: } 0 \le \beta_{m,n} \le 1, \forall m, n, \\
& \quad \text{C6: } \sum_{m\in\mathcal{M}} \beta_{m,n} \le 1, \forall n, \\
& \quad \text{C7: } b_{m,l} t_{m,l} \le T_c.
\end{aligned} \tag{24}$$

We set the objective optimization problem P (i.e., the total system cost) as Cost, where $\bar{\omega}$ is the weighting factor, $\bar{\omega} = (0, 1]$. Constraint C1 denotes that the offloading decision for each UD is a binary variable, constraint C2 indicates that only one offloading method can be selected per UD, constraints C3 and C4 stipulate that the computational resources allocated to the UDs do not exceed the inherent limitations of the UAVs or satellites, C5 and C6 ensure that the total bandwidth allocated by the UAV to the UD does not exceed its available bandwidth, and constraint C7 ensures that the UD task offloading time to the satellite is within its coverage time.

Since the optimization objectives $T$ and $E$ have different magnitudes and units, the two conflicting objectives are normalized, which can provide a complete Pareto optimal solution for our problem [12]. We introduce a weighting factor $\bar{\omega}$ to balance the objective functions:

$$\begin{aligned}
\text{P0: } & \min_{a,b,\beta,f} \sum_{m\in\mathcal{M}} \left\{ \bar{\omega} \frac{T_m - T_{\min}}{T_{\max} - T_{\min}} + (1-\bar{\omega}) \frac{E_m - E_{\min}}{E_{\max} - E_{\min}} \right\} \\
& \text{s.t. C1–C8.}
\end{aligned} \tag{25}$$

Here, $T_{\max}$ and $T_{\min}$ represent the maximum and minimum values of the total system computation delay, respectively, while $E_{\max}$ and $E_{\min}$ represent the maximum and minimum values of the total system computation delay for the reformulated problem P0. We can obtain the following formula:

$$\begin{aligned}
\text{P0: } & \min_{a,b,\beta,f} \sum_{m\in\mathcal{M}} \{\bar{\omega}AT_m + (1-\bar{\omega})BE_m\} \\
& \text{s.t. C1–C8.}
\end{aligned} \tag{26}$$

The solution value of problem P0 is denoted by $Q$, where $V = \bar{\omega}\frac{T_{\min}}{T_{\max}-T_{\min}} + (1-\bar{\omega})\frac{E_{\min}}{E_{\max}-E_{\min}}$, $A = \frac{1}{T_{\max}-T_{\min}}$, $B = \frac{1}{E_{\max}-E_{\min}}$ are constants and $\bar{\omega}$ is the weighting factor $\bar{\omega} = (0, 1]$.

The numerical value of the optimization objective depends on the transfer rate of the user task, computational resource allocation, and user offload scheduling. Owing to their closer proximity to UDs compared with satellites, UAVs can serve multiple users simultaneously. Furthermore, efficient bandwidth allocation can further reduce transmission costs. Moreover, given the limited computational resources of UAVs and satellites, efficient resource allocation for UDs is essential to minimize computational costs. Finally, to realize space-air-ground integrated networking, user terminals must dynamically integrate UAVs and satellites by jointly scheduling their complementary resources, thereby minimizing the total cost for all users. However, optimizing offload scheduling is complex due to its binary nature, affecting both bandwidth and resource allocation, with these variables being interdependent. The non-convexity of the optimization objective P and constraints C1 and C2 makes direct solving difficult. In this paper, we decompose the optimization problem and provide solutions for the subproblems discussed in Section 4.

## 4. Proposed solution

Based on the above analysis, the optimization objective is a mixed-integer nonlinear programming (MINLP) problem, which is difficult to solve directly. We decompose it into three subproblems: bandwidth allocation, computational resource allocation, and task offload scheduling, and solve them by convex optimization, the Lagrange multiplier method, and ADMM algorithms, respectively. Finally, an AOTORA algorithm is proposed jointly for the three subproblems to minimize the total user computational cost.

### 4.1. Bandwidth allocation optimization

Given the parameters $a$ and $b$, the bandwidth and computational resources optimization problem is not coupled; therefore, this paper is divided into two subsections, 4.1 and 4.2, to explain the solution process. Subsection 4.1 focuses on optimizing the bandwidth allocation problem. Through the observation and analysis, we find that P1 is an LP problem, where the meaning of $\beta_{m,n}$ is a portion of the bandwidth between UDs and UAVs. The optimization problem can be expressed as

$$\begin{aligned}
\text{P1:}\quad & \min_{\beta} \varphi \\
\text{s.t.}\ & \sum_{m\in\mathcal{M}} \bar{\omega}\left(\frac{D_m}{\beta_{m,n}B^n\log_2(1+\chi)}\right) + (1-\bar{\omega})\left(p^m\cdot\frac{D_m}{\beta_{m,n}B^n\log_2(1+\chi)}\right) \le \varphi \\
& 0 \le \beta_{m,n} \le 1, \forall m, n, \\
& \sum_{m\in\mathcal{M}} \beta_{m,n} \le 1, \forall n,
\end{aligned} \tag{27}$$

where $\mathcal{X} = \frac{H_m^n p_{m,n}}{\sum_{m\neq i} H_i^n p_{i,n}+\sigma^2}$. By taking the second derivative of the optimization variable $\beta_{m,n}$ in P1, we can obtain $\frac{\partial^2(\varphi)}{\partial(\beta_{m,n})^2} > 0$, so P1 is a convex function, which is addressed using the convex optimization algorithm tool, the CVX toolkit [9], to obtain $\beta^*$.

### 4.2. Optimization of computing resource allocation

Given the parameters $a$, $b$ and the bandwidth allocation $\beta$, the optimization problem with respect to the computational resource $f$ can be formulated as

$$\begin{aligned}
\text{P2.1:}\ & \min_{f_{m,n}} \sum_{m\in\mathcal{M}} \left[\bar{\omega}\left(\frac{\lambda_m}{f_{m,n}}\right) + (1-\bar{\omega})\left(p^n\cdot\frac{\lambda_m}{f_{m,n}}\right)\right] \\
\text{s.t.}\ & 0 \le f_{m,n} \le F_n^{\max} \\
& \sum_{m\in\mathcal{M}} f_{m,n} \le F_n^{\max}.
\end{aligned} \tag{28}$$

$$\begin{aligned}
\text{P2.2:}\ & \min_{f_{m,l}} \sum_{m\in\mathcal{M}} \left[\bar{\omega}\left(\frac{\lambda_m}{f_{m,l}}\right) + (1-\bar{\omega})\left(p^l\cdot\frac{\lambda_m}{f_{m,l}}\right)\right] \\
\text{s.t.}\ & 0 \le f_{m,l} \le F_l^{\max} \\
& \sum_{m\in\mathcal{M}} f_{m,l} \le F_l^{\max}.
\end{aligned} \tag{29}$$

We set the objective function of Eq. (28) as $\varphi_1$, and we obtain

$$\begin{aligned}
& \frac{\partial(\varphi_1)}{\partial(f_{m,n})} = -\bar{\omega}\frac{\lambda_m}{f_{m,n}} - (1-\bar{\omega})\frac{p^m\cdot\lambda_m}{f_{m,n}} < 0, \\
& \frac{\partial^2(\varphi_1)}{\partial(f_{m,n})^2} > 0, \\
& \frac{\partial^2(\varphi_1)}{\partial(f_{m,n})\partial(f_{m',n})} = 0, \\
& 0 \le m \le M,\ m \neq m'.
\end{aligned} \tag{30}$$

We observe that the objective function $\varphi_1$ of problem P2.1 is positive definite and can be addressed using the Lagrange multiplier method (similarly for P2.2). The Lagrange functions are expressed as

$$L_1(f_{m,n}, \mu) = \sum_{m\in\mathcal{M}} \left[\bar{\omega}\left(\frac{\lambda_m}{f_{m,n}}\right) + (1-\bar{\omega})\left(p^n \cdot \frac{\lambda_m}{f_{m,n}}\right)\right] + \mu\left(\sum_{m\in M} f_{m,n} - F_n^{\max}\right), \tag{31}$$

$$L_2(f_{m,l}, \nu) = \sum_{m\in\mathcal{M}} \left[\bar{\omega}\left(\frac{\lambda_m}{f_{m,l}}\right) + (1-\bar{\omega})\left(p^l \cdot \frac{\lambda_m}{f_{m,l}}\right)\right] + \nu\left(\sum_{m\in M} f_{m,l} - F_l^{\max}\right), \tag{32}$$

where $\mu$ and $\nu$ are both non-negative, and the Karush–Kuhn–Tucker (KKT) conditions for Eq. (31) are formulated as

$$\frac{\partial L(f_{m,n}, \mu)}{\partial f_{m,n}} = -\left[\bar{\omega}\frac{\lambda_m}{(f_{m,n})^2} + (1-\bar{\omega})\frac{p^n \cdot \lambda_m}{(f_{m,n})^2}\right] + \mu = 0, \tag{33}$$

$$\frac{\partial L(f_{m,n}, \mu)}{\partial \mu} = \sum_{m\in\mathcal{M}} f_{m,n} - F_n^{\max} = 0, \tag{34}$$

$$\mu\left(\sum_{m\in\mathcal{M}} f_{m,n} - F_n^{\max}\right) = 0. \tag{35}$$

From the above equations, we obtain

$$\sqrt{\mu} = \sum_{m\in\mathcal{M}} \frac{\sqrt{\bar{\omega}\cdot\lambda_m + (1-\bar{\omega})\cdot\lambda_m\cdot p^n}}{F_n^{\max}}. \tag{36}$$

Therefore, the optimal solution ${f_{m,n}}^*$ can be expressed as

$${f_{m,n}}^* = \frac{\lambda_m(\bar{\omega} + (1-\bar{\omega})\cdot p^n)}{\sum_{m\in\mathcal{M}}\sqrt{\lambda_m(\bar{\omega} + (1-\bar{\omega})\cdot p^n)}} \cdot F_n^{\max}. \tag{37}$$

Similarly, according to the KKT conditions of Eq. (32), the optimal solution ${f_{m,l}}^*$ can be expressed as

$${f_{m,l}}^* = \frac{\lambda_m(\bar{\omega} + (1-\bar{\omega})\cdot p^l)}{\sum_{m\in M}\sqrt{\lambda_m(\bar{\omega} + (1-\bar{\omega})\cdot p^l)}} \cdot F_l^{\max}. \tag{38}$$

### 4.3. User scheduling optimization

Given $\beta$ and $f$ found in subsections 4.1 and 4.2 above, the optimization problem for the offloading decisions $a$ and $b$ is described as

$$\begin{aligned} &\text{P3: } \min_{a,b} \sum_{m\in\mathcal{M}} \left\{\bar{\omega}T_m + (1-\bar{\omega})E_m\right\} \\ &\text{s.t. C1–C8.} \end{aligned} \tag{39}$$

Since $a$ and $b$ are binary variables, integer programming algorithms such as genetic algorithms have slower search speeds and require more training time, while the computation load of the cutting plane method is large, which increases the complexity of problem solving to some extent [2,11,42]. Taking these factors into account, this paper introduces a distributed computation offloading algorithm based on ADMM to solve the problem. However, this algorithm needs to evaluate the feasible region of each variable, introducing linear relaxation and transforming the original problem P3 into a convex optimization problem. The feasible regions of the optimization variables $a$ and $b$, $\wedge_a$ and $\wedge_b$, are defined as follows:

$$\wedge = \begin{Bmatrix} \wedge_a \\ \wedge_b \end{Bmatrix} \left| \begin{array}{l} \sum_{m\in\mathcal{M}} a_{m,n} f_{m,n} \le F_n^{\max}, 0 \le a_{m,n} \le 1, \forall m, n. \\ \sum_{m\in\mathcal{M}} b_{m,l} f_{m,l} \le F_l^{\max}, 0 \le b_{m,l} \le 1, \forall m, l. \\ b_{m,l} t_{m,l} \le T_c, \forall m, l. \end{array} \right\}. \tag{40}$$

The original question P3 can be rewritten as

$$\begin{aligned} \text{P3}'\text{:} \min_{a,b} & \sum_{m\in\mathcal{M}} \left\{ \bar{\omega} T_m + (1-\bar{\omega}) E_m \right\} \\ \text{s.t.} \quad & \text{Eq. (45), C2, C8.} \end{aligned} \tag{41}$$

The augmented Lagrange function is as follows:

$$\begin{aligned} L_\rho(a,b) = \text{P3}'(a_{m,n}, b_{m,l}) + \gamma^T \left( \sum_{n\in\mathcal{N}} a_{m,n} + \sum_{l\in\mathcal{L}} b_{m,l} - \mathbf{1} \right) \\ + \frac{\rho}{2} \left( \sum_{n\in\mathcal{N}} a_{m,n} + \sum_{l\in\mathcal{L}} b_{m,l} - \mathbf{1} \right)^2, \end{aligned} \tag{42}$$

where $\gamma$ is the Lagrange multiplier vector, $\rho$ is the penalty parameter, and $\mathbf{1}$ is the $M$-dimensional vector. This objective function and constraints are convex; $a$, $b$, and $\gamma$ are updated in sequential order as

$$\begin{aligned} a^{(i+1)} = \arg\min_a \Bigg[ \text{P3}' \left( a_{m,n}, {b_{m,l}}^{(i)} \right) + (\gamma^{(i)})^T \left( \sum_{n\in\mathcal{N}} a_{m,n} + \sum_{l\in\mathcal{L}} {b_{m,l}}^{(i)} - \mathbf{1} \right) \\ + \frac{\rho}{2} \left( \sum_{n\in\mathcal{N}} a_{m,n} + \sum_{l\in\mathcal{L}} {b_{m,l}}^{(i)} - \mathbf{1} \right)^2 \Bigg], \end{aligned} \tag{43}$$

$$\begin{aligned} b^{(i+1)} = \arg\min_b \Bigg[ \text{P3}' \left( {a_{m,n}}^{(i+1)}, b_{m,l} \right) + (\gamma^{(i)})^T \left( \sum_{n\in\mathcal{N}} {a_{m,n}}^{(i+1)} + \sum_{l\in\mathcal{L}} b_{m,l} - \mathbf{1} \right) \\ + \frac{\rho}{2} \left( \sum_{n\in\mathcal{N}} {a_{m,n}}^{(i+1)} + \sum_{l\in\mathcal{L}} b_{m,l} - \mathbf{1} \right)^2 \Bigg], \end{aligned} \tag{44}$$

$$\begin{aligned} \gamma^{(i+1)} = \arg\min_\gamma \Bigg[ \text{P3}' \left( {a_{m,n}}^{(i+1)}, {b_{m,l}}^{(i+1)} \right) + (\gamma)^T \left( \sum_{n\in\mathcal{N}} {a_{m,n}}^{(i+1)} + \sum_{l\in\mathcal{L}} {b_{m,l}}^{(i+1)} - \mathbf{1} \right) \\ + \frac{\rho}{2} \left( \sum_{n\in\mathcal{N}} {a_{m,n}}^{(i+1)} + \sum_{l\in\mathcal{L}} {b_{m,l}}^{(i+1)} - \mathbf{1} \right)^2 \Bigg]. \end{aligned} \tag{45}$$

The termination condition, defined by $\left\| a^{(i+1)} - a^{(i)} \right\| \le \varepsilon_{d2}$, $\left\| b^{(i+1)} - b^{(i)} \right\| \le \varepsilon_{d1}$, is satisfied when the algorithm stops or reaches the limit of iterations $I$. In this section, the original problem is solved by transforming the discrete variables into continuous variables so that the nonlinear problem P3 is turned into a linear problem and an explanation is given. Now, the continuous variable is reduced to a binary variable taking values $a$ and $b$. The specific recovery method is as follows:

$$a^* = \begin{cases} 1, & \text{if } a_{m,n} = \max\{a_{m,n}, b_{m,l}\}, \forall m, n, l \\ 0, & \text{otherwise} \end{cases}. \tag{46}$$

We denote the binary variable as $\wedge^* = \{a^*, b^*\}$. The method for solving P3 is detailed in Algorithm 1.

**Algorithm 1.** ADMM for solving P3.

**Input:** $M, N, \varepsilon_{d2}, \varepsilon_{d1}$.
**Initialize:** $a^{(0)}, b^{(0)}, f^{(0)} = \left\{{f_{m,n}}^{(0)}, {f_{m,l}}^{(0)}\right\}, \beta^{(0)}, i = 0$.
**Repeat:**
    update the global variable $a^{(i+1)}$ according to Eq. (43),
    update the global variable $b^{(i+1)}$ according to Eq. (44),
    update the global variable $\gamma^{(i+1)}$ according to $a^{(i+1)}$, $b^{(i+1)}$, and Eq. (45).
$i = i + 1$
**until** $||a^{(i+1)} - a^{(i)}||_2 \leq \varepsilon_{d2}$, $||b^{(i+1)} - b^{(i)}||_2 \leq \varepsilon_{d1}$.
**Output** the optimized solutions $a^*$ and $b^*$.

### 4.4. Overall algorithm convergence and complexity analysis

The algorithm iteratively solves three subproblems, refining the solution at each step. By introducing relaxation and approximation, the iterative updates converge to near-optimal results. Algorithm 2 outlines the iterative process for solving the overall optimization problem P. In each iteration, when we fix two of the variable blocks, the remaining single subproblem is usually convex or can be transformed into a convex problem. This means that solving each subproblem strictly reduces the total system cost. Since there is a lower bound on the total cost determined by system constraints, the cost sequence generated by the algorithm is monotonically decreasing and bounded below, which guarantees that the algorithm will converge to a stable point. Similar alternating schemes have also been shown to converge efficiently in existing literature [24,36]. Algorithm 2 outlines the iterative process for solving the overall optimization problem P. A more detailed convergence proof is as follows:

First, in the $r$-th iteration, for a given $\{\wedge^r = \{a^r, b^r\}\}$ , we obtain

$$Q\left(\beta^{(r)}, f^{(r)}, \Lambda^{(r)}\right) \geq Q\left(\beta^{(r+1)}, f^{(r)}, \Lambda^{(r)}\right). \tag{47}$$

Second, for a given $\{\beta^{(r)}, \wedge^r = \{a^r, b^r\}\}$, by obtaining the optimal computing CPU frequency through Eqs (37) and (38), we get

$$Q\left(\beta^{(r+1)}, f^{(r)}, \Lambda^{(r)}\right) \geq Q\left(\beta^{(r+1)}, f^{(r+1)}, \Lambda^{(r)}\right). \tag{48}$$

Third, in the ADMM-based Algorithm 1, the objective value $L$ of problem P3 decreases monotonically with the iteration index $i$:

$$L_\rho\left(a^{(i)}, b^{(i)}\right) \geq L_\rho\left(a^{(i+1)}, b^{(i+1)}\right), \tag{49}$$

$$L_\rho\left(a^{(i+1)}, b^{(i)}\right) \geq L_\rho\left(a^{(i+1)}, b^{(i+1)}\right), \tag{50}$$

$$L_\rho\left(a^{(i)}, b^{(i+1)}\right) \geq L_\rho\left(a^{(i+1)}, b^{(i+1)}\right). \tag{51}$$

**Algorithm 2.** AOTORA for solving P.

**Input:** $\varepsilon, r = 0, \beta^{(0)}, f^{(0)} = \left\{{f_{m,n}}^{(0)}, {f_{m,l}}^{(0)},\right\}, \wedge^{(0)} = \left\{a^{(0)}, b^{(0)}\right\}$.
**Repeat:**
    solve for $\beta^{(r+1)}$via the CVX toolbox with given $a^{(r)}$ and $b^{(r)}$,
    solve for $f^{(r+1)}$ via the Lagrange multiplier method with given $a^{(r)}$ and $b^{(r)}$,
    solve for $\wedge^{(r+1)}$ by Algorithm 1.
$r = r + 1$
**until** $|Q^{(r+1)}(\beta, f, \wedge) - Q^{(r+1)}(\beta, f, \wedge)|_2 \leq \varepsilon$.
Binary variables $a$ and $b$ recovery.
**Output** the optimized solutions ${\beta^{(r+1)}}^*$,${f^{(r+1)}}^*$, ${\wedge^{(r+1)}}^*$.

Therefore, for a given $\{\beta^{(r+1)}, f^{(r+1)} = \{f_{m,n}{}^{(r+1)}, f_{m,l}{}^{(r+1)}\}\}$, it has the following relationship:

$$Q\left(\beta^{(r+1)}, f^{(r+1)}, \Lambda^{(r)}\right) \geq Q\left(\beta^{(r+1)}, f^{(r+1)}, \Lambda^{(r+1)}\right). \tag{52}$$

Based on the above analysis, we can conclude that

$$Q\left(\beta^{(r)}, f^{(r)}, \Lambda^{(r)}\right) \geq Q\left(\beta^{(r+1)}, f^{(r+1)}, \Lambda^{(r+1)}\right). \tag{53}$$

The convergence of Algorithm 2 is guaranteed by the following two points: on the one hand, its objective function value $E$ decreases monotonically with the number of iterations $r$; on the other hand, the algorithm always keeps the solution within the feasible domain that satisfies all IoT device constraints. Therefore, it can be proven that Algorithm 2 has convergence properties.

Next, we evaluate the complexity of the AOTORA algorithm. We use the CVX toolbox to solve problem P1, and its computational complexity is $O(MN)$. We solve problem P2 using the KKT conditions obtained through the Lagrange multiplier method, with the complexity denoted as $O(2M + N + 1)$. In problem P3, the optimization involves $3M$ variables and $3M(N + 1)$ constraints, and its complexity is $O(3M(3M(N + 1)))$. In summary, the full complexity per iteration of the AOTORA algorithm is $O(MN+2M+N+1+3M(3M(N+1)))$, and the overall algorithm complexity is $R \cdot O(MN + 2M + N + 1 + 3M(3M(N + 1))) = R \cdot O(M^2)$, where $R$ represents the final iteration count of Algorithm 2.

Finally, the alternating optimization (AO) framework adopted in this paper is theoretically based on the block coordinate descent method [1]. The core idea of this principle is to approximate the optimal solution of the original problem by decomposing a complex joint optimization problem into several easier-to-solve subproblems and sequentially optimizing individual blocks of variables while keeping the others fixed. Compared with ESAO algorithm of Xie et al. [39], AO achieves directional development through systematic variable rotation, whereas ESAO relies on random mutations for global exploration. The two algorithms are thus suitable for problems with a clear structure and a black-box complexity, respectively. This study is further inspired by Liu et al.'s [20,21] predict-then-optimize hybrid architecture. In the future, combining the AO framework with time-series prediction models is planned to enable a shift from passive response to proactive planning, promoting the evolution of air-space-ground network optimization toward anticipatory decision-making.

## 5. Performance evaluation

### 5.1. Simulation setup

We examine the performance of the alternating iteration algorithm with guaranteed convergence by setting up an experimental environment, proving the effectiveness and convergence of the proposed AOTORA algorithm through simulation, and comparing it with other benchmark algorithms in different environments to evaluate its performance. The other primary parameters involved in the experiments are shown in Table 2 [3,7,16,18,37].

**Table 2.** System parameters

| Parameter | Value | Parameter | Value |
|---|---|---|---|
| $M$ | 30 | $B^n$ | 40 MHz |
| $N$ | 9 | $B^l$ | 5 GHz |
| $h$ | 50 m | $F_n^{\max}$ | 3 Gcycles/s |
| $\kappa$ | 2 | $F_l^{\max}$ | 8 Gcycles/s |
| $D_m$ | [3,6] MB | $p^m$ | 1 W |
| $\lambda_m$ | [100,300] Mcycle | $p^n$ | 5 W |
| $\gamma_0$ | –60 | $p^l$ | 10 W |
| $\theta$ | 25 C° | $h_{m,l}$ | 700 km |

In this paper, MATLAB is employed for simulation. Three LEO satellites are flying an area of 1000 m × 1000 m at an altitude of 700 km with the orbit altitude of $h_{m,l}$ = 700 km. The distance between UD $m$ and LEO satellite $l$ is $d_{m,l}$. $M$ UDs and $N$ UAVs are randomly placed in the predetermined region. Assuming that UAVs are hovering at a set altitude $h$, the channel power gain $\gamma_0$ at a reference distance of 1 m is set to –60 dB, the path loss exponent for the UD–UAV link is $\kappa = 2$, each UD transmits with a power of 1 W, whereas the UAV and the satellite transmit with powers of 5 W and 10 W, respectively. The noise power is $10^{-13}$ W. The task size and computational cycle requirements for each UD are randomly set within a fixed interval.

### 5.2. Simulation results and performance analysis

The article compares the algorithm proposed in this paper with four benchmark algorithms, including random optimization (RO) [5], centralized offloading scheme (COS) [5], equal bandwidth allocation (EBA) [43], offloading low-earth-orbit satellite (OLS) [40], and distributionally robust optimization (DRO) [35]. RO means that the ground user randomly offloads self-generated tasks to the UAV or satellite with equal probability. In the COS scheme, the ground-based cloud data center serves as the central coordinator of the inputs, and the interior point method is used to solve the optimization problem. EBA means that all users perform computational offloading with the UAV, providing equal bandwidth communication services. OLS means that all UDs will offload their own generated tasks to the satellite. DRO introduces a robust optimization baseline with a box uncertainty set [44], adjusting the key model parameter, channel gain $H_m^n$ and $H_m^l$, to fluctuate up and down by ±10%.

Figure 2a illustrates the relationship between the normalized system cost $Q$ and the weighting factor $\bar{\omega}$ under different weights and various user tasks. It can be observed that as the delay weighting factor increases, the $Q$ value also increases. Since the optimization objectives of delay and energy consumption conflict with each other, a normalization method is used for the solution. Regardless of how the weighting factor changes, the optimized value $Q$ remains within a certain range and does not fluctuate significantly.

Figure 2b analyzes the convergence and sensitivity to initial conditions of the AOTORA algorithm proposed in this paper. Simulation results show that regardless of changes in the weighting factors, the proposed algorithm consistently decreases rapidly during the early iterations and stabilizes after the fifth iteration. In addition, we exchanged the iteration order of the three subproblems in 4.1, 4.2, and 4.3 (hereafter denoted as A, B, and C, respectively), and the experimental results indicate that the final results remain stable, with fluctuations of the optimization problem P value Cost staying within 1%. In Fig. 3a, we present a sensitivity analysis of the key penalty parameter $\rho$ on the system's AOTORA algorithm. The experiments show that, as the penalty parameter $\rho$ increases, the convergence speed accelerates, but the overall final Cost value remains essentially unchanged. The above results demonstrate its robustness and also indicate that the proposed AOTORA algorithm has good convergence performance.

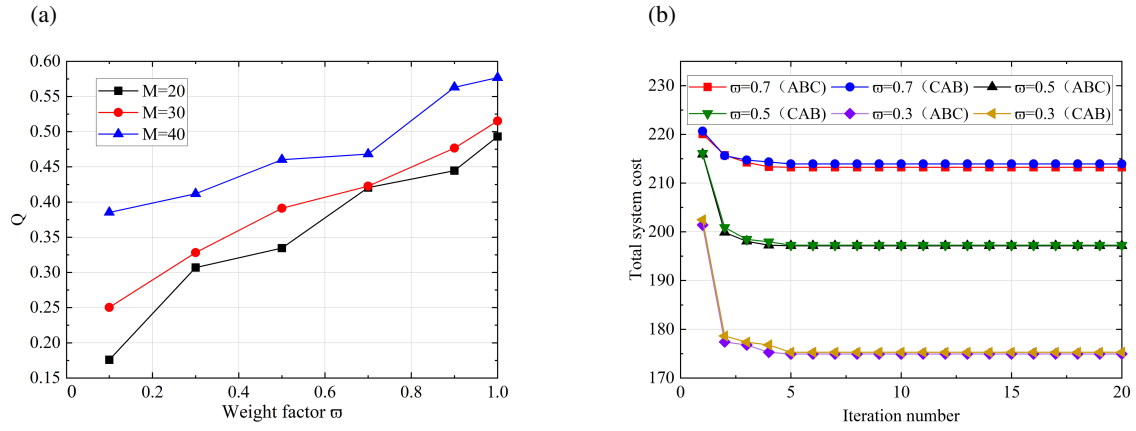


**Fig. 2.** Performance of AOTORA algorithm: comparison of $Q$ values for different $\bar{\omega}$ and different device quantities (a) and sensitivity analysis of AOTORA algorithm performance to different weight factors and iteration orders (b).

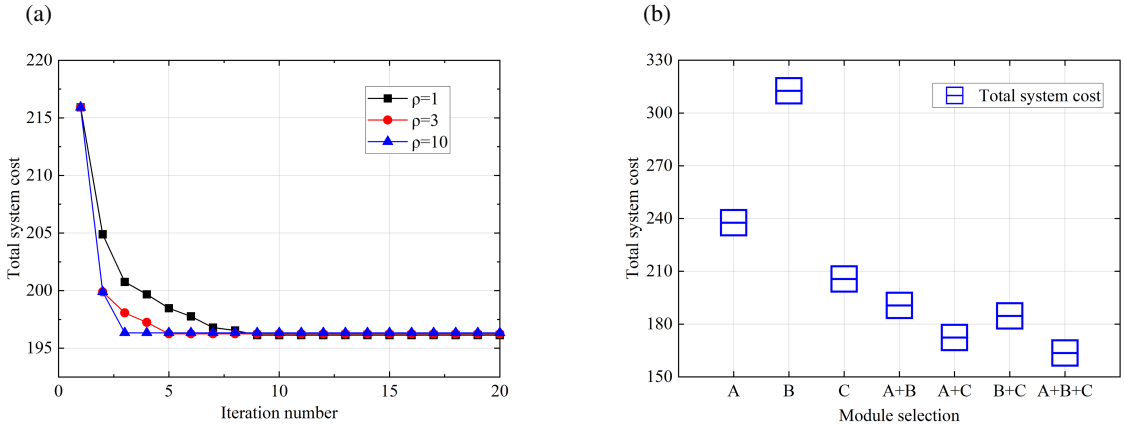


**Fig. 3.** Performance of AOTORA algorithm: sensitivity analysis of the system with respect to the penalty factor $\rho$ (a) and ablation study comparison of each subproblem and its combination modules (b).

Figure 3b evaluates the optimization contributions of each subproblem and its combined modules through an ablation study. The experimental results show that when only a single module participates in the optimization, subproblem C has the largest contribution, indicating that offloading scheduling has a significant impact on the delay and energy consumption of computation offloading. Next, when two submodules participate together, their contributions are higher than that of a single module, with subproblems A and C contributing more, suggesting that bandwidth allocation has a secondary influence on the optimization objective. Finally, when all three submodules participate in the optimization, the optimization value is minimized, and the contribution is maximized, indicating that computing resources are the third influencing factor.

Figure 4a describes the relationship between the number of UDs in different algorithms and the total system cost. It is easy to see that as the number of UDs increases, the total system computation cost also becomes higher. First, the RO algorithm has a larger total computation cost because the system's allocation for bandwidth or offloading scheduling is probabilistic and random. Second, although the EBA algorithm uses ADMM to seek an optimal offloading schedule, communication between users and UAVs uses equal bandwidth, which, to some extent, increases the total system cost. In contrast, for the OLS algorithm, the calculation offloading cost for satellites is higher than that for UAVs due to distance factors. For the DRO algorithm, channel fluctuations degrade link quality, reducing computation task transmission, and the system needs to invoke more computational resources to compensate, ultimately leading to a higher total system cost. Therefore, based on the above analysis and simulation results, the proposed AOTORA scheme reduces the average cost by 25.8% compared with the RO algorithm, by 12.48% compared with the OLS algorithm, by 11.63% compared with the EBA algorithm, and by 6.48% compared with the DRO algorithm. Compared with other methods, the AOTORA scheme shows slower growth in system computation cost and can effectively reduce the total cost of computation tasks.

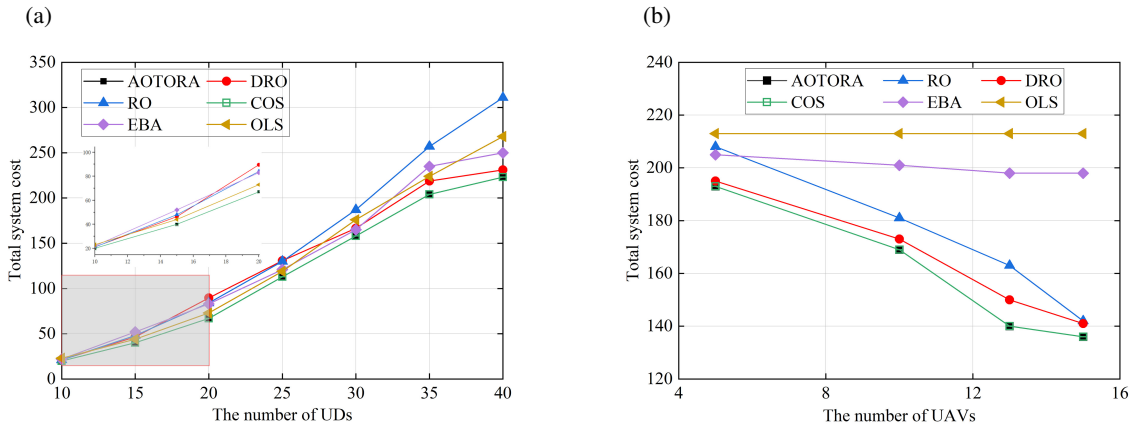


**Fig. 4.** Performance of AOTORA algorithm: relationship between number of devices and total system cost (a) and relationship between UAV quantity and total system cost (b).

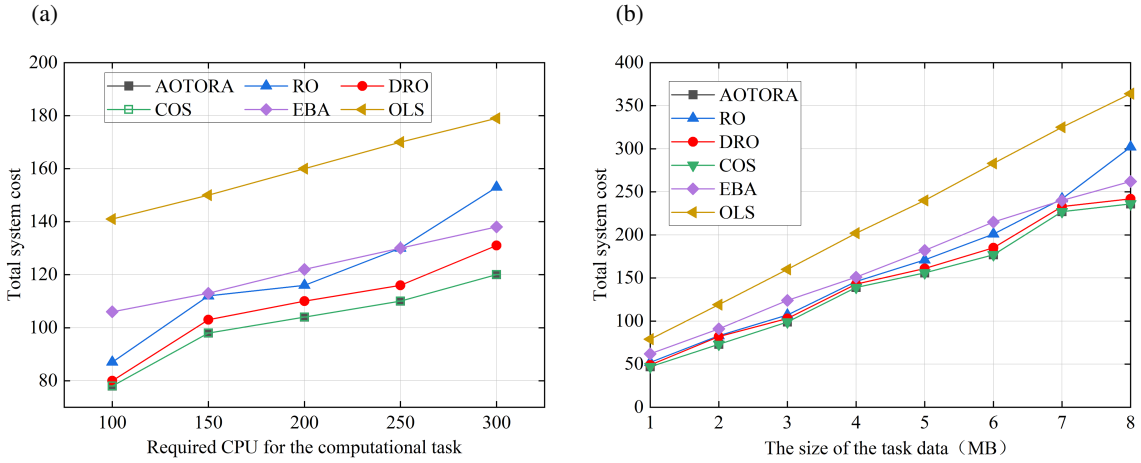


**Fig. 5.** Performance of AOTORA algorithm: comparison of task CPU frequency and total system cost (a) and comparison of task size and total system cost (b).

Figure 4b describes the relationship between the number of UAVs and the total system cost under different algorithms. We can observe that the AOTORA algorithm outperforms the RO, OLS, and EBA algorithms. First, as the number of UAVs increases, the total system cost decreases. This is because with more UAVs, the computational resources in the system increase, allowing more tasks to be offloaded to UAVs for processing, thereby reducing the total system cost. The channel fluctuations in the DRO algorithm can increase the overhead to some extent. However, there is a certain cost associated with devices processing tasks, so when the number of UAVs reaches 15, the total system costs of the AOTORA, COS, RO, and DRO algorithms are relatively close. Second, due to the average allocation of bandwidth in the EBA algorithm, the cost difference of processing tasks on UAVs and satellites is small, so an increase in the number of UAVs has little impact on the algorithm's performance. Finally, since tasks are not offloaded to UAVs in the OLS algorithm, increasing the number of UAVs does not affect the algorithm's performance.

Figure 5a illustrates the relationship between the number of CPU cycles required for computing tasks and the total system cost under different algorithms. We can observe that as the number of CPU cycles required by computing tasks increases, the total system computing cost for each algorithm also rises. Specifically, the OLS algorithm has higher computing costs than the other algorithms due to distance factors; the simulation results for the RO algorithm fluctuate more because of its randomness, and the DRO algorithm shows variations due to channel fluctuations, sometimes falling below or exceeding the EBA algorithm. Overall, compared with the RO, OLS, and EBA algorithms, the AOTORA algorithm proposed in this paper demonstrates better performance and can effectively reduce the total system computing cost.

Figure 5b describes the relationship between system task data size and total system cost. As shown in Fig. 5b, the proposed AOTORA algorithm is compared with other schemes. It can be observed that as the task scale increases, the total system cost increases in all five optimization algorithms. This is because, for ground users, whether offloading tasks to UAVs or satellite devices, the total system cost is related to the task scale and increases as the task scale grows. However, as the data volume increases, compared to the random allocation and scheduling in RO, the idle UAV computing resources in OLS, and the equal-bandwidth transmission in EDA, the proposed AOTORA algorithm jointly optimizes bandwidth allocation, computing resource allocation, and user scheduling, more effectively reducing the total computation cost for users. In addition, the curve of the proposed AOTORA algorithm is almost identical to that of the COS algorithm, indicating that the proposed computation offloading strategy can achieve the optimal solution of the COS algorithm.

Figure 6 illustrates the total computing costs of five optimization algorithms (AOTORA, RO, OLS, EBA, and DRO) and the distribution of total system costs for different algorithms. It can be seen from the figure that for the RO and EBA algorithms, a larger portion of users' total computing costs is distributed on LEO satellites, while the OLS algorithm offloads all user tasks to LEO satellites, which undoubtedly increases the system's total cost. In the DRO algorithm, the offloading proportion between UAVs and satellites is balanced, whereas the AOTORA algorithm offloads a

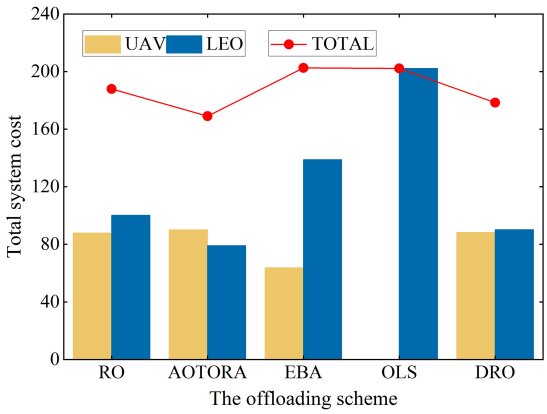


**Fig. 6.** Distribution of computing costs under different mechanisms.

larger proportion of tasks to UAVs, resulting in lower system overhead. Its total cost is lower than that of the other four algorithms because UAVs, compared to satellites, have certain computing resources and lower costs, playing a key role in this system.

### 5.3. Applicability and adaptability analysis

The AOTORA algorithm achieves effective control of computational complexity by decomposing complex problems into independent subproblems, such as resource allocation and task scheduling. Simulations show that its solution time grows approximately linearly with the number of users, rather than exponentially. This means that whether in small local networks or large-scale networks covering wide areas, the algorithm can complete optimization within an acceptable time, possessing the ability to cope with dynamic network expansion and contraction.

Although the algorithm relies on perfect, static channel conditions, it operates under periodic constraints. Before the start of a cycle in the joint optimization framework, when the quality of a link (such as a satellite link) is monitored, the algorithm selects high-quality channels, thereby adjusting task offloading decisions and resource allocation schemes to maintain overall performance. The algorithm's optimization objective – a weighted sum of delay and energy consumption – serves as a flexible regulator itself. The network can directly adjust the weights according to the priority of services in different scenarios. For example, in emergency rescue, low latency is prioritized, while in routine monitoring, low energy consumption is prioritized, meeting the diverse needs of real-world applications.

## 6. Conclusion and future work

This paper investigates the problem of remote IoT task offloading and resource allocation for LEO satellites under a dual-layer heterogeneous network collaborative architecture, considering their high mobility and dynamically changing channel environment, with the goal of minimizing the total computation cost for UDs. To address the complex MINLP problem, the optimization problem is decomposed into different subproblems, which are solved individually. An AOTORA algorithm based on the space-air dual-layer network architecture is proposed. Simulation results show that the proposed scheme outperforms other benchmark schemes in both small-scale and large-scale scenarios. In terms of the average total computation cost for users, it reduces costs by 12.48%, 25.8%, 11.63%, and 6.84% compared with OLS, RO, EBA, and DRO, respectively. Despite these achievements, this work is based on a static resource allocation framework within a single time slot and does not account for the dynamic randomness of tasks and the ultra-large-scale networks characteristic of IoT scenarios. Therefore, in our future work, we will study a multi-time-slot dynamic optimization framework that incorporates task data dependencies and user mobility models to enhance long-term computation adaptability.

### Data availability statement

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

**Acknowledgments**

This work was supported by the Key Project of the Henan Province Science and Technology Research and Development Program Joint Fund (grants No 235200810040 and 231111221500), the Major Science and Technology Projects of the Longmen Laboratory (grant No. 231100220500), the National Natural Science Foundation of China (grant No. 62072158), and in part by the Henan Province Science and Technology Project (grant No. 242102210192). The publication costs of this article were partially covered by the Estonian Academy of Sciences.

## References

1. Bertsekas, D. *Convex Optimization Theory*. Athena Scientific, Belmont, Massachusetts, 2009.
2. Boyd, S. and Vandenberghe, L. Localization and cutting-plane methods. *Stanford EE 364b Lecture Notes*, 2007, **386**, 2.
3. Chen, B., Li, N., Li, Y., Tao, X. and Sun, G. Energy efficient hybrid offloading in space-air-ground integrated networks. In *2022 IEEE Wireless Communications and Networking Conference (WCNC), Austin, TX, USA, 10–13 April 2022*. IEEE, 2022, 1319–1324. https://doi.org/10.1109/WCNC51071.2022.9771798
4. Chen, J.-H., Kuo, W.-C. and Liao, W. SpaceEdge: optimizing service latency and sustainability for space-centric task offloading in LEO satellite networks. *IEEE Transactions on Wireless Communications*, 2024, **23**(10), 15435–15446. https://doi.org/10.1109/TWC.2024.3429510
5. Chen, X., Zhang, H., Wu, C., Mao, S., Ji, Y. and Bennis, M. Optimized computation offloading performance in virtual edge computing systems via deep reinforcement learning. *IEEE Internet of Things Journal*, 2019, **6**(3), 4005–4018. https://doi.org/10.1109/JIOT.2018.2876279
6. Chen, Y., Ai, B., Niu, Y., Zhang, H. and Han, Z. Energy-constrained computation offloading in space-air-ground integrated networks using distributionally robust optimization. *IEEE Transactions on Vehicular Technology*, 2021, **70**(11), 12113–12125. https://doi.org/10.1109/TVT.2021.3116593
7. Chen, Z., Zhou, H., Du, S., Liu, J., Zhang, L. and Liu, Q. Reinforcement learning-based resource allocation and energy efficiency optimization for a space-air-ground-integrated network. *Electronics*, 2024, **13**(9), 1792. https://doi.org/10.3390/electronics13091792
8. Cheng, N., Lyu, F., Quan, W., Zhou, C., He, H., Shi, W. et al. Space/aerial-assisted computing offloading for IoT applications: a learning-based approach. *IEEE Journal on Selected Areas in Communications*, 2019, **37**(5), 1117–1129. https://doi.org/10.1109/JSAC.2019.2906789
9. Grant, M., Boyd, S. and Ye, Y. *CVX: Matlab software for disciplined convex programming*. Stanford University, Stanford, CA, USA, 2008.
10. He, L., Li, J., Wang, Y., Zheng, J. and He, L. Balancing total energy consumption and mean makespan in data offloading for space-air-ground integrated networks. *IEEE Transactions on Mobile Computing*, 2024, **23**(1), 209–222. https://doi.org/10.1109/TMC.2022.3222848
11. Jiang, H., Lee, Y. T., Song, Z. and Wong, S. C.-w. An improved cutting plane method for convex optimization, convex-concave games, and its applications. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing,* 2020, 944–953. https://doi.org/10.1145/3357713.3384284
12. Kers, J., Majak, J., Goljandin, D., Gregor, A., Malmstein, M. and Vilsaar, K. Extremes of apparent and tap densities of recovered GFRP filler materials. *Composite Structures*, 2010, **92**(9), 2097–2101. https://doi.org/10.1016/j.compstruct.2009.10.003
13. Kong, L., Tan, J., Huang, J., Chen, G., Wang, S., Jin, X. et al. Edge-computing-driven Internet of Things: a survey. *ACM Computing Surveys*, 2022, **55**(8), 1–41. https://doi.org/10.1145/3555308
14. Laghari, A. A., Wu, K., Laghari, R. A., Ali, M. and Khan, A. A. Retracted article: A review and state of art of Internet of Things (IoT). *Archives of Computational Methods in Engineering*, 2022, **29**(3), 1395–1413. https://doi.org/10.1007/s11831-021-09622-6
15. Li, J., Shi, Y., Dai, C., Yi, C., Yang, Y., Zhai, X. et al. A learning-based stochastic game for energy efficient optimization of UAV trajectory and task offloading in space/aerial edge computing. *IEEE Transactions on Vehicular Technology*, 2025, **74**(6), 9717–9733. https://doi.org/10.1109/TVT.2025.3540964
16. Li, S., Li, W., Zheng, W., Xia, Y., Guo, K., Peng, Q. et al. Multi-user joint task offloading and resource allocation based on mobile edge computing in mining scenarios. *Scientific Reports*, 2025, **15**, 16170. https://doi.org/10.1038/s41598-025-00730-y
17. Li, W., Li, S., Hao, J., Wu, Q. and Wang, R. Efficient task offloading and resource allocation in space-air-ground-sea networks: a MAPPO-based approach. In *Wireless Artificial Intelligent Computing Systems and Applications. WASA 2025. Lecture Notes in Computer Science* (Cai, Z., Zhu, Y., Wang, Y. and Qiu, M., eds). Springer, Singapore, 2025, **15688**, 117–126. https://doi.org/10.1007/978-981-96-8731-2_12
18. Liao, Z., Ma, Y., Huang, J., Wang, J. and Wang, J. HOTSPOT: a UAV-assisted dynamic mobility-aware offloading for mobile-edge computing in 3-D space. *IEEE Internet of Things Journal*, 2021, **8**(13), 10940–10952. https://doi.org/10.1109/JIOT.2021.3051214
19. Liu, J., Shi, Y., Fadlullah, Z. M. and Kato, N. Space-air-ground integrated network: a survey. *IEEE Communications Surveys & Tutorials*, 2018, **20**(4), 2714–2741. https://doi.org/10.1109/COMST.2018.2841996
20. Liu, J., Hou, Z., Liu, B. and Zhou, X. Mathematical and machine learning innovations for power systems: predicting transformer oil temperature with beluga whale optimization-based hybrid neural networks. *Mathematics*, 2025, **13**(11), 1785. https://doi.org/10.3390/math13111785
21. Liu, J., Hou, Z., Wang, B. and Yin, T. Optimizing microgrid energy management via DE-HHO hybrid metaheuristics. *Computers, Materials & Continua*, 2025, **84**(3). https://doi.org/10.32604/cmc.2025.066138
22. Liu, Y., Jiang, L., Qi, Q. and Xie, S. Energy-efficient space–air–ground integrated edge computing for Internet of Remote Things: a federated DRL approach. *IEEE Internet of Things Journal*, 2023, **10**(6), 4845–4856. https://doi.org/10.1109/JIOT.2022.3220677
23. Mach, P. and Becvar, Z. Mobile edge computing: a survey on architecture and computation offloading. *IEEE Communications Surveys & Tutorials*, 2017, **19**(3), 1628–1656. https://doi.org/10.1109/COMST.2017.2682318
24. Mao, S., He, S. and Wu, J. Joint UAV position optimization and resource scheduling in space-air-ground integrated networks with mixed cloud-edge computing. *IEEE Systems Journal*, 2021, **15**(3), 3992–4002. https://doi.org/10.1109/JSYST.2020.3041706
25. Maray, M. and Shuja, J. [Retracted] Computation offloading in mobile cloud computing and mobile edge computing: survey, taxonomy, and open issues. *Mobile Information Systems*, 2022, **2022**(1), 1121822. https://doi.org/10.1155/2022/1121822
26. Mei, C., Gao, C., Wang, H., Xing, Y., Ju, N. and Hu, B. Joint task offloading and resource allocation for space–air–ground col-

laborative network. *Drones*, 2023, **7**(7), 482. https://doi.org/10.3390/drones7070482
27. Qi, X., Chong, J., Zhang, Q. and Yang, Z. Collaborative computation offloading in the multi-UAV fleeted mobile edge computing network via connected dominating set. *IEEE Transactions on Vehicular Technology*, 2022, **71**(10), 10832–10848. https://doi.org/10.1109/TVT.2022.3188554
28. Ren, C., Zhang, G., Gu, X. and Li, Y. Computing offloading in vehicular edge computing networks: full or partial offloading?. In *2022 IEEE 6th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 4–6 March 2022*. IEEE, 2022, 693–698. https://doi.org/10.1109/ITOEC53115.2022.9734360
29. Sadeeq, M. M., Abdulkareem, N. M., Zeebaree, S. R., Ahmed, D. M., Sami, A. S. and Zebari, R. R. IoT and cloud computing issues, challenges and opportunities: a review. *Qubahan Academic Journal*, 2021, **1**(2), 1–7. https://doi.org/10.48161/qaj.v1n2a36
30. Satyanarayanan, M., Bahl, P., Caceres, R. and Davies, N. The case for VM-based cloudlets in mobile computing. *IEEE Pervasive Computing*, 2009, **8**(4), 14–23. https://doi.org/10.1109/MPRV.2009.82
31. Shang, B., Yi, Y. and Liu, L. Computing over space-air-ground integrated networks: challenges and opportunities. *IEEE Network*, 2021, **35**(4), 302–309. https://doi.org/10.1109/MNET.011.2000567
32. Shen, S., Shen, G., Dai, Z., Zhang, K., Kong, X. and Li, J. Asynchronous federated deep-reinforcement-learning-based dependency task offloading for UAV-assisted vehicular networks. *IEEE Internet of Things Journal*, 2024, **11**(19), 31561–31574.
33. Sriharsha, C. and Murthy, C. S. R. Energy-efficient computation offloading in 6G space-air-ground integrated networks. In *2022 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), Gandhinagar, Gujarat, India, 18–21 December 2022*. IEEE, 2023, 1–6. https://doi.org/10.1109/ANTS56424.2022.10227801
34. Su, J., Yu, S., Li, B. and Ye, Y. Distributed and collective intelligence for computation offloading in aerial edge networks. *IEEE Transactions on Intelligent Transportation Systems*, 2023, **24**(7), 7516–7526. https://doi.org/10.1109/TITS.2022.3160594
35. Sun, J., Chen, X., Jiang, C. and Guo, S. Distributionally robust optimization of on-orbit resource scheduling for remote sensing in space-air-ground integrated 6G networks. *IEEE Journal on Selected Areas in Communications*, 2025, **43**(1), 382–395. https://doi.org/10.1109/JSAC.2024.3460057
36. Tang, Q., Fei, Z., Li, B. and Han, Z. Computation offloading in LEO satellite networks with hybrid cloud and edge computing. *IEEE Internet of Things Journal*, 2021, **8**(11), 9164–9176. https://doi.org/10.1109/JIOT.2021.3056569
37. Wang, N., Li, F., Chen, D., Liu, L. and Bao, Z. NOMA-based energy-efficiency optimization for UAV enabled space-air-ground integrated relay networks. *IEEE Transactions on Vehicular Technology*, 2022, **71**(4), 4129–4141. https://doi.org/10.1109/TVT.2022.3151369
38. Wang, X., Chen, H. and Tan, F. Hybrid OMA/NOMA mode selection and resource allocation in space-air-ground integrated networks. *IEEE Transactions on Vehicular Technology*, 2024. https://doi.org/10.1109/TVT.2024.3452477
39. Xie, J., He, J., Gao, Z., Wang, S., Liu, J. and Fan, H. An enhanced snow ablation optimizer for UAV swarm path planning and engineering design problems. *Heliyon*, 2024, **10**(18, e37819). https://doi.org/10.1016/j.heliyon.2024.e37819
40. Xu, S., Li, X., Zhang, J., Li, F., Liang, Y. and Hao, S. Bilateral collaborative computing offloading via LEO satellites for remote network applications. *Computer Networks*, 2025, **261**, 111124. https://doi.org/10.1016/j.comnet.2025.111124
41. Zhang, J., Zhang, J., Shen, F., Yan, F. and Bu, Z. DOGS: dynamic task offloading in space-air-ground integrated networks with game-theoretic stochastic learning. *IEEE Internet of Things Journal*, 2025, **12**(2), 1655–1672. https://doi.org/10.1109/JIOT.2024.3457855
42. Zhang, L., Xie, Y., Chen, J., Feng, L., Chen, C. and Liu, K. A study on multiform multi-objective evolutionary optimization. *Memetic Computing*, 2021, **13**(3), 307–318. https://doi.org/10.1007/s12293-021-00331-y
43. Zhang, Y., Na, Z., Wen, Z., Nallanathan, A. and Lu, W. Joint service caching, computation offloading and resource allocation for dual-layer aerial Internet of Things. *Computer Networks*, 2025, **257**, 110974. https://doi.org/10.1016/j.comnet.2024.110974
44. Zhong, L., Liu, Y., Deng, X., Wu, C., Liu, S. and Yang, L. T. Distributed optimization of multi-role UAV functionality switching and trajectory for security task offloading in UAV-assisted MEC. *IEEE Transactions on Vehicular Technology*, 2024, **73**(12), 19432–19447. https://doi.org/10.1109/TVT.2024.3434354

## Ühiste tööülesannete allalaadimine ja ressursijaotus kaug-IoT jaoks kosmos-õhk-maa võrkude arhitektuuris

**Zhenhua Li, Guoqiang Zheng, Zhe Han, Pingjie Xia, Huahong Ma ja Baofeng Ji**

Piirkondades, kus maismaataristu on ebapiisav, on kasutajaseadmetel piiratud arvutusressursid, mistõttu lühikese viivituse ja energiatõhusa andmetöötluse tagamine osutub keeruliseks. Selle probleemi lahendamiseks pakub käesolev töö välja kahekihilise heterogeense võrguarhitektuuri, mis kasutab mehitamata õhusõidukite ja Maa-lähedase orbiidi (LEO) satelliitide arvutusressursse. Arvestades LEO-satelliitide suurt liikuvust, kanali muutlikkuse parameetreid ja tööülesannete allalaadimise järjekorra viivitust, formuleeritakse optimeerimisülesanne mittelineaarse segatüüpi täisarvulise programmeerimise probleemina. Eesmärk on minimeerida viivituse ja energiatarbimise kaalutud summat (st kogu süsteemi kulu). Töös pakutakse välja arvutuslikult lihtne alternatiivne algoritm. Algne probleem jaotatakse kolmeks alamprobleemiks: ribalaiuse jaotus, protsessori taktsageduse jaotus ja ülesannete ajastamise optimeerimine. Need alamülesanded lahendatakse vastavalt kumeroptimeerimise, Lagrange'i kordajate meetodi ja vahelduvate suundadega Lagrange'i kordajate meetodi abil. Lõpuks rakendatakse Pareto optimaalset jaotust parima tasakaalu leidmiseks erinevate optimeerimiskriteeriumide vahel. Arvutustulemused näitavad, et väljatöötatud AOTORA süsteemi keskmine kogukulu väheneb vastavalt 25,8%, 11,63%, 12,48% ja 6,84% võrreldes stohhastilise optimeerimise, võrdse ribalaiuse jaotuse, LEO-allalaadimisalgoritmi ja jaotuslikult robustse optimeerimisega.