Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

# Resource Allocation and Task Offloading in Blockchain-Enabled UAV-Assisted LEO Satellite Edge Computing Using Dual Decomposition

**Aziz Bahmani**

aziz.bahmani@uok.ac.ir

University of Kurdistan

**Mohammad Fathi**

University of Kurdistan

**Article**

**Keywords:** MEC, blockchain, optimization, task offloading, UAVs, LEOs, resource allocation, Lagrangian

**Posted Date:** November 24th, 2026

**DOI:** https://doi.org/10.21203/rs.3.rs-11687739/v1

**License:** This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

**Additional Declarations:** No competing interests reported.

# Resource Allocation and Task Offloading in Blockchain-Enabled UAV-Assisted LEO Satellite Edge Computing Using Dual Decomposition

Aziz Bahmani[1] and Mohammad Fathi[1]

[1]Department of Electrical and Computer Engineering, University of Kurdistan, Sanandaj, Iran.

Contributing authors: aziz.bahmani@uok.ac.ir; mfathi@uok.ac.ir;

**Abstract**

This paper investigates a secure and energy-efficient three-tier mobile edge computing (MEC) architecture composed of mobile terminals (MTs), unmanned aerial vehicles (UAVs), and a low earth orbit (LEO) satellite. MTs offload computation tasks to nearby UAVs for local execution or forwarding to the LEO satellite when local resources are insufficient. To ensure data integrity and secure coordination, a blockchain-enabled framework is deployed between the UAV and LEO layers. A unified system model is developed to characterize communication and computation delays, energy consumption, and bandwidth and CPU constraints. The joint task offloading and resource allocation problem is formulated as a mixed-integer nonlinear programming problem with the objective of minimizing total system energy under latency and blockchain generation constraints. To address the computational complexity, Lagrangian relaxation and dual decomposition are applied to decompose the problem into UAV-level and LEO-blockchain subproblems. A low-complexity joint iterative resource allocation and user association algorithm with closed-form solutions is derived. Simulation results demonstrate notable energy saving compared with geometric-programming-based benchmarks.

**Keywords:** MEC, blockchain, optimization, task offloading, UAVs, LEOs, resource allocation, Lagrangian

# 1 Introduction

Mobile edge computing (MEC) integrating airborne (UAV-based) and space (LEO satellite) nodes enables lower latency and wider service coverage for mobile terminal (MT) applications, particularly in infrastructure-limited or remote areas. However, offloading decisions must trade off delay, energy, computation capacity, and communication constraints. Moreover, secure and auditable metadata about offloading decisions and accounting is critical. Numerous studies have investigated edge computing in 5G and 6G environments. Recently, the combination of UAV and LEO networks has been highlighted for providing coverage in infrastructure-free areas. Similarly, blockchain has gained attention for enhancing security and privacy in IoT and aerospace systems. However, research addressing a unified UAV–LEO–Blockchain integrated model remains limited, since most works handle only partial components.

In [1], a multi-UAV two-tier MEC network with blockchain was studied, but no satellite or IoT layer is modeled. In [2], a three-layer LEO–UAV–IoT system and a blockchain sub-system are introduced; task partitioning is non-binary, simplifying computations. Their DDPG-based learning framework requires long training time, in contrast to our analytical approach using analytical continuous-resource updates with heuristic binary task assignment and explicit delay–energy trade-offs. In [4], a LEO–UAV–IoT model was analyzed focusing on maximizing the number of offloaded tasks and trajectory planning, using game theory and reinforcement learning, yet omitting blockchain and per-task computation frequency optimization. The study in [5] investigated three-layer task splitting but enabled non-binary divisions and neglected satellite CPU and bandwidth constraints as well as blockchain. Work [6] used alternating direction method of multipliers (ADMM) relaxation for a two-layer device–satellite–cloud model focusing only on energy minimization.

Reference [8] modeled a device–satellite system considering non-binary splitting and interference control via iterative convex optimization, ignoring bandwidth constraints. Study [9] proposed UAVs jointly performing computation and blockchain consensus but excluded satellites and bandwidth coupling. In [10] a three-layer emergency LEO–UAV–MT system was optimized through DDPG; tasks were processed solely in satellites without UAV computational role or blockchain. In [11], UAVs served as edge servers with blockchain security, sequentially optimizing bandwidth, CPU frequency, and blockchain subproblems — single-layer only and no LEO cooperation. The study in [12] presented a device–UAV–mobile-site structure integrated with blockchain and ADMM-based CPU/bandwidth optimization, yet ignored individual task latency constraints and feasibility.

In [14], the joint energy consumption of user equipments and a single UAV was minimized via a Lagrangian-based approach, where each task was divided into partial segments completed within the total flight duration. However, that model lacked binary offloading and blockchain mechanisms. In [13], the total energy of a MEC system and blockchain delay was minimized by jointly optimizing transmit power, processing frequency, and user association. Their analytical closed-form solution, though elegant, excluded bandwidth constraints and a third-layer extension. In [15], the authors minimized the overall energy and delay of a single-UAV MEC network using successive convex approximation (SCA) for bandwidth, computing frequency, and task

partition optimization, yet neglected blockchain and per-task latency limits. In [16], total energy consumption encompassing computation, blockchain, and delay penalties was minimized by optimizing CPU frequency and UAV trajectory, but without considering task-specific delay or bandwidth coupling.
In [17], instead of energy, total power consumption was minimized via joint optimization of user equipment and UAV transmit powers, UAV positions, and frequency distribution—again without bandwidth or blockchain integration. Finally, [18] minimized total user latency through joint optimization of bandwidth and frequency allocation under node energy constraints, but similarly omitted blockchain and per-task delay enforcement.
Overall, only a limited number of existing studies jointly address energy minimization in a three-tier LEO–UAV–IoT architecture that integrates blockchain-based security while fully coupling bandwidth, CPU-frequency, and per-task delay constraints. This gap motivates the integrated optimization framework proposed in this paper. Specifically, we impose explicit maximum tolerable delay constraints for both task execution and blockchain block generation while enabling processing among multiple UAVs.

The main contributions of this work are summarized as follows:

1. **Three-tier architecture and problem formulation:** We develop a blockchain-enabled MT–UAV–LEO MEC architecture and formulate a joint task offloading and resource allocation problem to minimize total system energy subject to bandwidth, CPU, and maximum tolerable delay constraints for both task execution and block generation.
2. **Dual decomposition optimization:** The resulting mixed-integer nonconvex problem is efficiently decomposed into dynamically coupled subproblems via Lagrangian and dual decomposition, enabling decoupled optimization across UAVs, the LEO satellite, and the blockchain subsystem.
3. **Low-complexity algorithm design:** Leveraging the analytical structure of the Lagrangian, we derive a low-complexity iterative algorithm with closed-form solutions for MT assignment, bandwidth allocation, and CPU-frequency allocation at both UAV and LEO layers, while satisfying per-task computational requirements

The remainder of this paper is organized as follows. Section II describes the three-tier MT–UAV–LEO system model, encompassing communication, computation, and blockchain components. Section III formulates the joint task offloading and resource allocation problem under energy, bandwidth, CPU, and delay constraints. Section IV derives closed-form solutions via Lagrangian relaxation and dual decomposition. Section V presents the proposed iterative assignment and resource allocation algorithm and describes its iterative implementation. Simulation results are reported in Section VI, and conclusions are drawn in Section VII.

## 2 System Model

We consider an space air ground system consisting of three layers as depicted in Fig. 1. In the first layer of this system, we have a set $\mathcal{M} \triangleq \{m : m = 1, \ldots, M\}$ of MTs, positioned in different parts of the area, which can be terminals or sensors based on

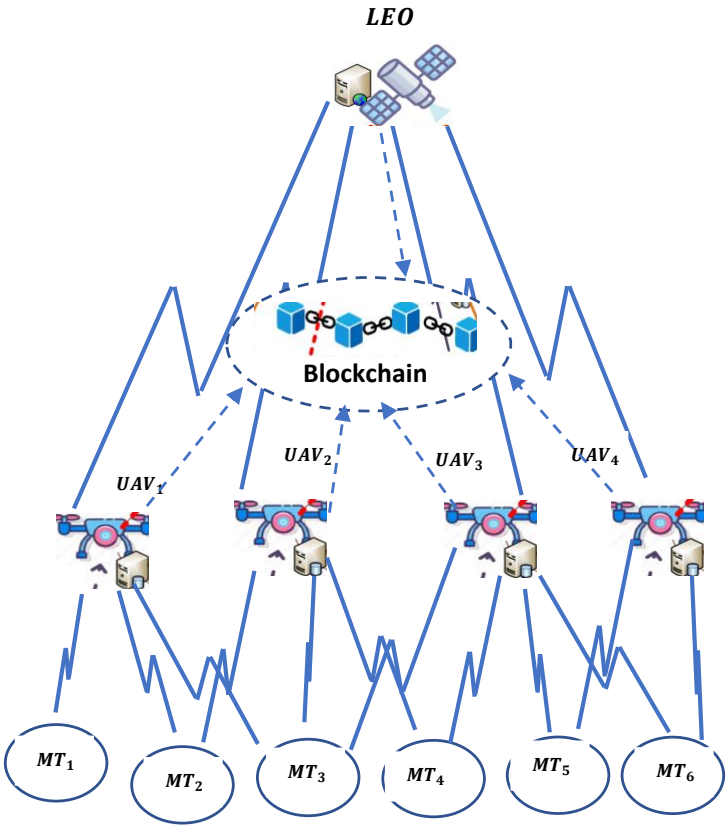



**Fig. 1**: System Model

the application. Each $\mathrm{MT}_m$ handles the offloading of a task to the system with a data size of $D_m$ bits. Moreover, the required CPU cycles per bit by a task of $\mathrm{MT}_m$ or its computation density is denoted by $C_m$ CPU cycles per bit.

In the second layer, we have a set of UAVs $\mathcal{U} \triangleq \{u : u = 1, \ldots, U\}$, that peform mobile edge computing for offloaded tasks from MTs, and UAVs themselves can have internal tasks that must be processed. As an example of UAV internal task, we can imagine that they take images of the area when they go pass by, for monitoring smart farming, pipeline, road or rail inspection and also periodic imaging of flooded or disaster-stricken areas in emergency scenarios. In the third and final layer, we have a LEO satellite, acting as a high computing edge/cloud processor. In alongside, LEO and UAVs act also as Blockchain network nodes to secure the metadata of tasks.

The defined parameters are summarized in Table 1, and the problem formulation is done in the following subsections.

**Table 1**: Summary of Notations

| Notation | Description |
|---|---|
| *Binary Variables and Sets* | |
| $\mathbf{X} = \{x_{m,u}\}$ | Binary assignment of MT $m$ offloading to UAV $u$ |
| $\mathbf{Z} = \{z_{m,u}\}$ | Binary execution of $\text{MT}_m$'s task at $\text{UAV}_u$ |
| $\mathbf{Y} = \{y_m^L\}$ | Binary offloading of $\text{MT}_m$'s task to LEO |
| $\boldsymbol{v}^L = \{v_u^L\}$ | Binary variable of $\text{UAV}_u$'s task offloads to LEO |
| $\mathbf{S} = \{s_u\}$ | Binary selection of $\text{UAV}_u$ as primary node |
| *Task and Computation Parameters* | |
| $D_m, D_u$ | Task data size of $\text{MT}_m$ / $\text{UAV}_u$ (bits) |
| $C_m, C_u$ | Comp. density (cycles/bit) of MT$m$/UAV$u$ task |
| $F_u, F_L$ | Comp. capacity of $\text{UAV}_u$ / LEO (cycles/s) |
| $f_{m,u}$ | CPU frequency at $\text{UAV}_u$ to $\text{MT}_m$ |
| $f_{u,0}$ | Local CPU frequency of $\text{UAV}_u$ |
| $f_m^L, f_u^L$ | CPU frequency at LEO to $\text{MT}_m$ / $\text{UAV}_u$ task |
| $f_b^L$ | CPU frequency at LEO for block generation |
| *Communication Parameters* | |
| $r_{m,u}$ | Rate of $\text{MT}_m$ to $\text{UAV}_u$ |
| $r_m^L, r_u^L$ | Rate of $\text{UAV}_u$ to LEO for $\text{MT}_m$ / $\text{UAV}_u$ task |
| $b_{m,u}$ | Bandwidth of $\text{MT}_m$ at $\text{UAV}_u$ |
| $b_m^L, b_u^L$ | Bandwidth of LEO to $\text{MT}_m$ / $\text{UAV}_u$ task |
| $B_u, B_L$ | Bandwidth capacity of $\text{UAV}_u$ / LEO |
| *Delay Parameters* | |
| $\tau_{m,U}^{\text{tr}}, \tau_{m,L}^{\text{tr}}$ | Transmission delay of MT $m$ to UAV $u$ / to LEO |
| $\tau_{u,L}^{\text{tr}}$ | Transmission delay of $\text{UAV}_u$ task to LEO |
| $\tau_{m,u}^{\text{cmp}}, \tau_{u,0}^{\text{cmp}}$ | Comp. delay of $\text{MT}_m$/$\text{UAV}_u$ task at $\text{UAV}_u$ |
| $\tau_{m,L}^{\text{cmp}}, \tau_{u,L}^{\text{cmp}}$ | Comp. delay of $\text{MT}_m$/$\text{UAV}_u$ task at LEO |
| $\tau_m^{\max}, \tau_u^{\max}$ | Maximum tolerable delay of $\text{MT}_m$ / $\text{UAV}_u$ |
| $\tau_b^{\max}$ | Maximum tolerable blockchain delay |
| *Energy Parameters* | |
| $E_{m,u}^{\text{tr}}, E_{m,L}^{\text{tr}}$ | Transmission energy of $\text{MT}_m$ to UAV / LEO |
| $E_{u,L}^{\text{tr}}$ | Transmission energy of $\text{UAV}_u$ task to LEO |
| $E_{m,u}^{\text{cmp}}, E_{u,0}^{\text{cmp}}$ | Comp. energy of $\text{MT}_m$/$\text{UAV}_u$ task at $\text{UAV}_u$ |
| $E_{m,L}^{\text{cmp}}, E_{u,L}^{\text{cmp}}$ | Comp. energy of $\text{MT}_m$/$\text{UAV}_u$ task at LEO |
| $E_b$ | Energy consumption of block generation at LEO |
| *Power Parameters* | |
| $p_m$ | Transmit power of $\text{MT}_m$ |
| $p_{u,m}$ | Relay power of $\text{UAV}_u$ for $\text{MT}_m$'s task to LEO |
| $p_{u,0}$ | Relay power of $\text{UAV}_u$ for its own task to LEO |
| $P_m, P_u$ | Maximum power constraint of $\text{MT}_m$/$\text{UAV}_u$ |
| $k_u, k_L$ | Effective capacitance coeff. of $\text{UAV}_u$/LEO |
| *Geometric and Channel Parameters* | |
| $d_{m,u}$ | Distance between $\text{MT}_m$ and $\text{UAV}_u$ |
| $d_u^L$ | Distance between $\text{UAV}_u$ and LEO |
| $(X_m, Y_m)$ | Coordinates of $\text{MT}_m$ |
| $(X_u, Y_u, H_u)$ | Coordinates and altitude of $\text{UAV}_u$ |
| $(X_L, Y_L, H_L)$ | Coordinates and altitude of LEO |
| $h_{m,u}, h_u^L$ | Channel gain of MT $m$ to UAV$u$/ UAV$u$ to LEO |
| $\sigma^2$ | Noise power spectral density |
| $\gamma$ | Path-loss exponent |
| *Blockchain Parameters* | |
| $I_b$ | Block size (bits) |
| $C_b$ | block generation Comp. intensity (cycles/bit) |
| $T_b^g$ | Block generation delay |
| $T_b^c$ | Consensus delay |
| $T_b^p$ | Propagation delay of UAV to LEO |
| $T_b$ | Total block creation delay |

## 2.1 Communication Model

### 2.1.1 MT to UAV link

The communication channel between MTs and UAVs is assumed to follow time-varying and flat-fading channels. To investigate the latency and energy consumption of UAV edge processing, the characteristics of the MT–UAV wireless communication link must be analyzed. The path loss between mobile terminal $\mathrm{MT}_m$ and $\mathrm{UAV}_u$ is modeled as [2]

$$\begin{aligned}\mathrm{PL}_{m,u} = 20\log_{10}\left(\frac{4\pi f_c}{c}\sqrt{d_{m,u}^2 + H_u^2}\right) \\ + \mathrm{pro}_{m,u}^{\mathrm{LoS}}\,\eta^{\mathrm{LoS}} + \left(1 - \mathrm{pro}_{m,u}^{\mathrm{LoS}}\right)\eta^{\mathrm{NLoS}}\end{aligned} \tag{1}$$

where $d_{m,u}$ denotes the horizontal distance between $\mathrm{MT}_m$ and $\mathrm{UAV}_u$, and $H_u$ represents the altitude of $\mathrm{UAV}_u$. The parameters $f_c$ and $c$ denote the carrier frequency and the speed of light, respectively. Moreover, $\eta^{\mathrm{LoS}}$ and $\eta^{\mathrm{NLoS}}$ denote the excessive path losses under line-of-sight (LoS) and non-line-of-sight (NLoS) conditions.

The probability of establishing a LoS link between $\mathrm{MT}_m$ and $\mathrm{UAV}_u$ is given by [2]

$$\mathrm{pro}_{m,u}^{\mathrm{LoS}} = \frac{1}{1 + \alpha_1 \exp\left(-\alpha_2\left[\arctan\left(\frac{H_u}{d_{m,u}}\right) - \alpha_1\right]\right)}, \tag{2}$$

where $\alpha_1$ and $\alpha_2$ are environment-dependent constants.

Based on the path loss in (1), the uplink transmit rate from $\mathrm{MT}_m$ to $\mathrm{UAV}_u$ is expressed as

$$r_{m,u} = b_{m,u}\log_2\left(1 + \frac{p_m\, h_{m,u}}{\sigma^2}\right), \tag{3}$$

where $h_{m,u} = 10^{-\mathrm{PL}_{m,u}/10}$ is the channel gain from $\mathrm{MT}_m$ to $\mathrm{UAV}_u$, $p_m$ is the transmit power of $\mathrm{MT}_m$, $b_{m,u}$ denotes the bandwidth allocated by $\mathrm{UAV}_u$ to $\mathrm{MT}_m$, and $\sigma^2$ is the noise power.

The assignment of $\mathrm{MT}_m$ to $\mathrm{UAV}_u$ is denoted by a binary variable $x_{m,u}$. If $\mathrm{MT}_m$ transmit its own task to $\mathrm{UAV}_u$ via communication with $\mathrm{UAV}_u$, then $x_{m,u} = 1$, otherwise $x_{m,u} = 0$. Since transmission for each MT is done only via one UAV, we enforce the constraint $\sum_{u=1}^{U} x_{m,u} = 1$. The transmitted tasks from MTs to UAVs can be executed in UAV or further relayed to the satellite due to insufficient processing power. We define $z_{m,u}$ to show the task execution of $\mathrm{MT}_m$ in $\mathrm{UAV}_u$ and $y_m^L$ to define task transmission of $\mathrm{MT}_m$ to LEO through any UAV. Therefore, the following constraints are always true for all $m$, $\sum_u z_{m,u} + y_m^L = 1$, $y_m^L z_{m,u} = 0$. Moreover, let's consider $B_u$ as the total bandwidth available at each UAV to be divided among assigned MTs. Considering $\Omega_u \triangleq \{m : x_{m,u} = 1\}$ as the set of MTs associated with $\mathrm{UAV}_u$, and where $b_{m,u}$ as the bandwidth asiigned to $\mathrm{MT}_m$, then $\sum_{m\in\Omega_u} b_{m,u}x_{m,u} = B_u$. If $D_m$ is the task data size of $\mathrm{MT}_m$ to be transmitted to $\mathrm{UAV}_u$, the delay and consumption energy

of this transmission are computed as

$$\tau_{m,U}^{tr} = \sum_{u} x_{m,u} \left( \frac{D_m}{r_{m,u}} \right) \tag{4}$$

$$E_{m,U}^{\mathrm{tr}} = p_m \tau_{m,U}^{tr} \tag{5}$$

respectively.

### 2.1.2 UAV to LEO Link

In the second communication layer, UAVs may offload tasks to the LEO satellite. Due to the high LEO altitude, UAV–LEO links are assumed to be quasi line-of-sight (LoS) with nearly identical path losses. Specifically, the path loss of $\mathrm{UAV}_u$ is modeled as $\mathrm{PL}_u^L = \mathrm{FSL} - G_t^{UAV} - G_r^{LEO} + \Delta_u$, where FSL is the free space loss, $G_t^{UAV}$ is the UAV transmit antenna gain, $G_r^{LEO}$ is LEO receive antenna gain and $\Delta_u \sim \mathcal{U}[0, \Delta_0]$ dB captures minor link variations across UAVs. Accordingly, the channel gain is given by $h_u^L = 10^{-\mathrm{PL}_u^L/10}$, and $\sigma^2$ denotes the noise power.

The binary variable $v_u^L$ indicates whether the internal task of $\mathrm{UAV}_u$ is offloaded to the LEO satellite ($v_u^L = 1$) or executed locally at the UAV ($v_u^L = 0$). Similarly, the binary variable $y_m^L$ denotes whether the task of $\mathrm{MT}_m$, after being received by $\mathrm{UAV}_u$, is relayed to the LEO satellite ($y_m^L = 1$).

Accordingly, the achievable uplink transmit rates from $\mathrm{UAV}_u$ to the LEO satellite are given by

$$r_m^L = b_m^L \, y_m^L \log_2 \left( 1 + \frac{p_{u,m}^L \, h_u^L / 10}{\sigma^2} \right), \tag{6}$$

$$r_u^L = b_u^L \, v_u^L \log_2 \left( 1 + \frac{p_{u,0}^L \, h_u^L}{\sigma^2} \right), \tag{7}$$

where $r_m^L$ and $r_u^L$ denote the transmit rates of the relayed task of $\mathrm{MT}_m$ and the internal task of $\mathrm{UAV}_u$, respectively. Moreover, $b_m^L$ and $b_u^L$ represent the bandwidth allocated by the LEO satellite to the relayed MT task and the UAV internal task, while $p_{u,m}^L$ and $p_{u,0}^L$ denote the corresponding transmit powers of $\mathrm{UAV}_u$.

Due to the UAV-to-LEO transmission, additional transmission delay and energy consumption are incurred, which are expressed as

$$\tau_{m,L}^{\mathrm{tr}} = y_m^L \left( \frac{D_m}{r_m^L} \right), \qquad \tau_{u,L}^{\mathrm{tr}} = v_u^L \left( \frac{D_u}{r_u^L} \right), \tag{8}$$

$$E_{m,L}^{\mathrm{tr}} = p_{u,m}^L \, \tau_{m,L}^{\mathrm{tr}}, \qquad E_{u,L}^{\mathrm{tr}} = p_{u,0}^L \, \tau_{u,L}^{\mathrm{tr}}, \tag{9}$$

where $D_m$ and $D_u$ denote the data sizes of the relayed task of $\mathrm{MT}_m$ and the internal task of $\mathrm{UAV}_u$, respectively.

## 2.2 Computation Model

Following the objective of task offloading in UAV or LEO, offloaded tasks to network are executed by leveraging their computation resources. Let's denote the total computation capacity at each $\text{UAV}_n$ and LEO as $F_u$ and $F_L$ in cycles per second. This capacity is allocated to offloaded tasks $m \in \Omega_n$ from MTs or UAVs internal tasks using predetermined rules to meet the offloading requirements. If $C_m$ and $C_u$ be the required CPU cycles per bit of $\text{MT}_m$ and $\text{UAV}_u$, then:

### 2.2.1 Task computation on UAV

Assuming $f_{m,u}$ and $f_{u,0}$ be the allocated computation frequency to $\text{MT}_m$ and $\text{UAV}_u$ internal task, the delay and consumption energy of these computations are calculated as

$$\begin{aligned} \tau_{m,U}^{cmp} &= \sum_{u=1}^{U} z_{m,u} \frac{C_m D_m}{f_{m,u}} \\ \tau_{u,0}^{cmp} &= (1 - v_u^L) \frac{C_u D_u}{f_{u,0}} \end{aligned} \tag{10}$$

and

$$\begin{aligned} E_{m,U}^{cmp} &= \sum_{u=1}^{U} z_{m,u} K_u C_m D_m f_{m,u}^2 \\ E_{u,0}^{cmp} &= (1 - v_u^L) k_u C_u D_u f_{u,0}^2 \end{aligned}$$

respectively, where $k_u$ is UAV's CPU effective capacitance coefficient, $v_u^L$ is local execution indicator for UAV u task, $z_{m,u}$ is task execution of terminal $m$ in UAV u indicator,

### 2.2.2 Task computation on LEO

when MT or UAVs task are relayed to LEO for processing, we have the following relations for delay and energy of execution:

$$\begin{aligned} \tau_{m,L}^{\text{cmp}} &= y_m^L \left( \frac{C_m D_m}{f_m^L} \right) \\ \tau_{u,L}^{\text{cmp}} &= v_u^L \left( \frac{C_u D_u}{f_u^L} \right) \end{aligned}$$

and

$$\begin{aligned} E_{m,L}^{\text{cmp}} &= y_m^L k_L \left( f_m^L \right)^2 C_m D_m \\ E_{u,L}^{\text{cmp}} &= v_u^L k_L \left( f_u^L \right)^2 C_u D_u \end{aligned}$$

where $k_L$ is LEO CPU effective capacitance, $f_u^L$ is computation frequency allocated to task $m$ in LEO, and $f_m^L$ computation frequency allocated to UAV$u$ in LEO.

## 2.3 Blockchain System

Besides the wireless network, a blockchain subsystem is used to ensure the security and integrity of all offloaded tasks. The LEO satellite and UAVs operate as blockchain nodes: they convert offloaded task data into transactions, group them into blocks, and then participate in block generation and consensus. Block generation aggregates all transactions into a block and is computationally intensive, while the consensus stage verifies the generated block across all nodes.

Following [2], we adopt a PBFT-based consensus mechanism tailored for space–air–ground integrated networks (SAGIN). The LEO, due to its large coverage and computational resources, acts as the block generator, while UAVs serve as consensus nodes. In PBFT, one UAV acts as the primary node and the remaining UAVs operate as non-primary nodes.

The SAGIN-tailored PBFT protocol consists of five steps: *request*, *preprepare*, *prepare*, *commit*, and *reply*. Only the computational costs relevant to delay evaluation are summarized below.

### Request

Each edge node generates and signs a transaction after task processing. The LEO packages all transactions into a signed and authenticated block and sends a request to the blockchain. The primary UAV verifies the block signature, MAC, and the signatures of all included transactions. If $\vartheta_s$ and $\vartheta_c$ denote the CPU cycles for signature and MAC verification, respectively, then the primary-node computation delay is

$$T_1 = \frac{(U+2)(\vartheta_s + \vartheta_c)}{f_u^{\mathrm{pr}}}. \tag{11}$$

### Preprepare

The primary UAV signs the block and generates $U-1$ MACs for the non-primary nodes, then broadcasts the preprepare message through the LEO. The corresponding delay is

$$T_{2,pr} = \frac{\vartheta_s + (U-1)\vartheta_c}{f_u^{\mathrm{pr}}}. \tag{12}$$

Each non-primary node verifies the received block and transactions, introducing additional delay

$$T_{2,\mathrm{npr}} = \max_u \frac{(U+2)(\vartheta_s + \vartheta_c)}{f_u^{\mathrm{npr}}}. \tag{13}$$

Thus, $T_2 = T_{2,pr} + T_{2,\mathrm{npr}}$.

### Prepare

After verifying the preprepare message, each non-primary node sends prepare messages to all nodes. Each node must collect at least $2\lceil F \rceil + 1$ consistent messages, where

$F = (U-1)/3$. The delays are

$$T_3^{\mathrm{pr}} = \frac{(2\lceil F \rceil + 1)(\vartheta_s + \vartheta_c)}{f_u^{\mathrm{pr}}}, \tag{14}$$

$$T_{3,u}^{\mathrm{npr}} = \frac{(2\lceil F \rceil + 1)(\vartheta_s + \vartheta_c) + \vartheta_s + (U-1)\vartheta_c}{f_u^{\mathrm{npr}}}, \tag{15}$$

and $T_3 = \max\{T_3^{\mathrm{pr}}, T_{3,u}^{\mathrm{npr}}\}$ for $u \in \mathcal{U}_{\mathrm{non}}$, where $\mathcal{U}_{\mathrm{non}}$ denotes the set of all non-primary nodes.

### Commit

Nodes broadcast commit messages and again must gather $2\lceil F \rceil + 1$ consistent commits. The CPU cycles required at each node are

$$\lambda_4 = \vartheta_s + (U-1)\vartheta_c + (2\lceil F \rceil + 1)(\vartheta_s + \vartheta_c). \tag{16}$$

The delay is

$$T_4 = \max\left\{ \frac{\lambda_4}{f^{\mathrm{pr}}}, \frac{\lambda_4}{f_u^{\mathrm{npr}}} \right\}. \tag{17}$$

### Reply

After gathering enough commit messages, each node stores the block and sends a reply to the LEO. Reply generation requires $\vartheta_s + \vartheta_c$ cycles:

$$T_5 = \max\left\{ \frac{\vartheta_s + \vartheta_c}{f^{\mathrm{pr}}}, \frac{\vartheta_s + \vartheta_c}{f^{\mathrm{npr}}} \right\}. \tag{18}$$

Since the steps do not overlap, the consensus delay is

$$T_b^c = T_1 + T_2 + T_3 + T_4 + T_5. \tag{19}$$

### Block Generation and Propagation

Let $I_b$ denote the block size and $C_b$ the block-generation intensity. If the LEO allocates $f_b^L$ cycles/s, then delay and energy of block generation are

$$T_b^g = \frac{C_b I_b}{f_b^L}, \quad E_b = k_L C_b I_b (f_b^L)^2. \tag{20}$$

Propagation delay $T_b^p$ is treated as a constant, giving the total block-creation delay

$$T_b = T_b^g + T_b^p + T_b^c. \tag{21}$$

Finally, a binary variable $s_u$ indicates the primary node, subject to $\sum_{u=1}^{U} s_u = 1$.

# 3 Problem Formulation

This section presents a problem formulation designed to minimize the total energy consumption of the integrated LEO–UAV–MT system, encompassing communication, computation, and blockchain processing. Transmission powers are not optimization variables. MT transmit powers are prescribed. For UAV–LEO transmission, a prescribed UAV power budget is shared among its currently offloaded tasks according to (43). For a fixed offloaded task set, these powers are held fixed during bandwidth and CPU allocation. The task power shares are recomputed if the offloaded task set changes.

This restriction simplifies selected continuous resource-allocation steps; it is not a relaxation that guarantees recovery of the optimum of a joint power-allocation problem.

The total energy to be minimized is expressed as

$$E_T = \sum_{m=1}^{M} \left(E_m^{\mathrm{tr}} + E_m^{\mathrm{cmp}}\right) + \sum_{u=1}^{U} \left(E_u^{\mathrm{tr}} + E_u^{\mathrm{cmp}}\right) + E_b + E_c \tag{22}$$

The first two summations account for transmission and computation energy of MT tasks and UAV internal tasks, respectively. The terms $E_b$ and $E_c$ represent LEO block-generation energy and UAV consensus computation energy. The latter is specified in (38).

Energy minimization is subject to communication and computation resource constraints together with delay requirements for task execution and block creation.

Each $\mathrm{MT}_m$ and $\mathrm{UAV}_u$ is associated with a maximum tolerable latency, $\tau_m^{\max}$ and $\tau_u^{\max}$, such that $t_m^{\mathrm{tr}} + t_m^{\mathrm{cmp}} \le \tau_m^{\max}$ and $t_u^{\mathrm{tr}} + t_u^{\mathrm{cmp}} \le \tau_u^{\max}$. Similarly, the blockchain subsystem must satisfy a block-generation delay limit, $T_b \le \tau_b^{\max}$.

For $\mathrm{MT}_m$ served by $\mathrm{UAV}_u$, the overall task latency (local or relayed via LEO) is given by

$$\begin{aligned} \tau_m^{\mathrm{tot}} = & \sum_{u=1}^{U} \left(x_{m,u} t_{m,u}^{\mathrm{tr}} + z_{m,u} t_{m,u}^{\mathrm{cmp}}\right) \\ & + y_m^L \left(\sum_{u=1}^{U} x_{m,u} t_{u,L}^{\mathrm{tr}} + t_{m,L}^{\mathrm{cmp}}\right). \end{aligned} \tag{23}$$

where the summation term accounts for the delay of transmission $\mathrm{MT}_m$'s task to UAV, and the delay of executing the task there, while the second term is when the task is uploaded to the LEO. For each UAV's local task,

$$\tau_u^{\mathrm{tot}} = (1 - v_u^L)\tau_{u,0}^{cmp} + v_u^L(\tau_{u,L}^{\mathrm{tr}} + \tau_{u,L}^{cmp})$$

The overall problem is then formulated as

$$\mathbf{(P)}: \quad \min_{X,Y,Z,\mathcal{F},\mathcal{B},s,\boldsymbol{v}^L} \; \min_{f_b^L, f_u^{\mathrm{npr}}, f_u^{\mathrm{pr}}} E_T,$$

subject to (for all $m, u$):

$$\tau_m^{\text{tot}} = \tau_m^{\text{tr}} + \tau_m^{\text{cmp}} \leq \tau_m^{\max} \tag{24a}$$
$$\tau_u^{\text{tot}} = \tau_u^{\text{tr}} + \tau_u^{\text{cmp}} \leq \tau_u^{\max} \tag{24b}$$
$$T_b \leq \tau_b^{\max} \tag{24c}$$
$$\sum_m b_{m,u} x_{m,u} \leq B_u \tag{24d}$$
$$\sum_m y_m^L b_m^L + \sum_u v_u^L b_u^L \leq B_L \tag{24e}$$
$$\sum_m y_m^L f_m^L + \sum_u v_u^L f_u^L + f_b^L \leq F_L \tag{24f}$$
$$\sum_{m=1}^{M} z_{m,u} f_{m,u} + (1 - v_u^L) f_{u,0} + f_u^{\text{pr}} s_u + f_u^{\text{npr}} (1 - s_u) \leq F_u \tag{24g}$$
$$\sum_{u=1}^{U} s_u = 1 \tag{24h}$$
$$\sum_{u=1}^{U} x_{m,u} = 1 \tag{24i}$$
$$z_{m,u} + y_m^L = x_{m,u} \tag{24j}$$
$$X, Y, Z, \boldsymbol{v}^L, \boldsymbol{s} \in \{0, 1\} \tag{24k}$$
$$f_{m,u}, f_{u,0}, f_m^L, f_u^L, f_b^L f_u^{\text{pr}}, f_u^{\text{npr}} b_{m,u}, b_m^L, b_u^L \geq 0. \tag{24l}$$

Here, $X$, $Y$, $Z$, $\boldsymbol{v}^L$, and $\mathbf{s}$ denote the binary assignment variables, while $\mathcal{F} = \{f_{m,u}\}$ and $\mathcal{B} = \{b_{m,u}\}$ represent CPU-frequency and bandwidth allocations at UAVs and LEO nodes. Description of all parameters are given in Table 1.

Problem **(P)** is a Mixed-Integer Nonlinear Program (MINLP), which is NP-hard and generally lacks an analytical solution. To handle its complexity, we next develop the Lagrangian relaxation and dual-decomposition framework.

# 4 Problem Solution

To address the coupling introduced by the delay constraints (24a) and (24b) in **(P)**, we introduce non-negative dual multipliers. The dualization is applied exclusively to the latency constraints associated with MT task completion and UAV internal processing, while all other constraints remain in their primal form. One dual variable is assigned per coupling constraint:

$$\lambda_m \geq 0 \quad (\text{MT}_m \text{ delay (24a)}),$$
$$\lambda_u \geq 0 \quad (\text{UAV}_u \text{ delay (24b)}).$$

The partial Lagrangian can thus be expressed as

$$\begin{aligned} L = &\sum_{m} \left(E_m^{\mathrm{tr}} + E_m^{\mathrm{cmp}}\right) + \sum_{u} \left(E_u^{\mathrm{tr}} + E_u^{\mathrm{cmp}}\right) + E_b + E_c \\ &+ \sum_{m} \lambda_m \left(\tau_m^{\mathrm{tr}} + \tau_m^{\mathrm{cmp}} - \tau_m^{\max}\right) \\ &+ \sum_{u} \lambda_u \left(\tau_u^{\mathrm{tr}} + \tau_u^{\mathrm{cmp}} - \tau_u^{\max}\right) \end{aligned} \tag{25}$$

The frequencies $f_u^{\mathrm{pr}}$ and $f_u^{\mathrm{npr}}$ are not free variables of $P_2$. They are fixed by the blockchain reservation in (38) and enter $P_2$ only through the residual CPU in (42).

Grouping the partial-Lagrangian terms motivates four sequential allocation stages. They remain coupled through association, offloading, and residual CPU capacities. Selected continuous subproblems are convex for fixed discrete decisions, but the overall procedure is heuristic, not an exact separable convex decomposition.

$$\min_{x,z,y,s,b,f,\ldots} \mathcal{L} = \underset{\text{MT Tra}}{\min} + \underset{\text{UAV Comp}}{\min} + \underset{\text{UAV Tra}}{\min} + \underset{\text{LEO Comp}}{\min}.$$

In the following subsections, each subproblem is presented along with its closed-form solution or specialized optimization procedure.

## 4.1 Subproblem (P1)

### 4.1.1 MT Association and UAV Bandwidth Allocation

The transmission component associated with MT–UAV links, weighted by the dual multiplier $\lambda_m$, is formulated as

$$P_1: \quad \min_{\{b_{m,u}, x_{m,u}\}} \sum_{m} \left(E_m^{\mathrm{tr}} + \lambda_m \tau_m^{\mathrm{tr}}\right) \tag{26}$$

Given that

$$\begin{aligned} E_m^{\mathrm{tr}} &= p_m \tau_m^{\mathrm{tr}}, \\ \tau_m^{\mathrm{tr}} &= \sum_{u} x_{m,u} \frac{D_m}{r_{m,u}} = \sum_{u} x_{m,u} \frac{D_m}{b_{m,u} \log_2\left(1 + h_{m,u} p_m / \sigma^2\right)} \end{aligned}$$

the relaxed optimization problem becomes

$$\begin{aligned} \min_{b_{m,u}, x_{m,u}} \quad & \sum_{m} \sum_{u} \frac{\alpha_{m,u}}{b_{m,u}} x_{m,u} \\ \text{s.t.} \quad & \sum_{m} b_{m,u} x_{m,u} \le B_u, \\ & \sum_{u} x_{m,u} = 1, \qquad x_{m,u} \in \{0,1\} \end{aligned} \tag{27}$$

where

$$\alpha_{m,u} \triangleq \frac{(p_m + \lambda_m) D_m}{\log_2\big(1 + h_{m,u} p_m / \sigma^2\big)}. \tag{28}$$

Let $\gamma_u$ denote the Lagrange multiplier associated with the UAV bandwidth constraint. The corresponding Lagrangian function is written as

$$\begin{aligned} L_1 &= \sum_m \sum_u \frac{\alpha_{m,u}}{b_{m,u}} x_{m,u} + \sum_u \gamma_u \left( \sum_m b_{m,u} x_{m,u} - B_u \right) \\ &= \sum_m \sum_u \left( \frac{\alpha_{m,u}}{b_{m,u}} + \gamma_u b_{m,u} \right) x_{m,u} - \sum_u \gamma_u B_u \end{aligned} \tag{29}$$

By applying the first-order optimality condition with respect to $b_{m,u}$,

$$\frac{\partial L_1}{\partial b_{m,u}} = -\frac{\alpha_{m,u}}{b_{m,u}^2} + \gamma_u = 0,$$

the optimal bandwidth allocation for $\mathrm{MT}_m$ at $\mathrm{UAV}_u$ is obtained as

$$b_{m,u}^* = \sqrt{\frac{\alpha_{m,u}}{\gamma_u}}. \tag{30}$$

By substituting the optimal bandwidth $b_{m,u}^*$ into the dualized objective in $L_1$, the optimal user association for each MT can be determined as

$$\begin{aligned} u_m^* &= \arg\min_u \left( \frac{\alpha_{m,u}}{b_{m,u}^*} + \gamma_u b_{m,u}^* \right) \\ &= \arg\min_u \; 2\sqrt{\alpha_{m,u} \gamma_u}. \end{aligned} \tag{31}$$

From the bandwidth capacity constraint $\sum_m b_{m,u} x_{m,u} = B_u$ and the optimal form of $b_{m,u}^*$, it follows that

$$\gamma_u = \left( \frac{\sum_{m \in \Omega_u} \sqrt{\alpha_{m,u}}}{B_u} \right)^2 \tag{32}$$

where $\Omega_u \triangleq \{ m \,|\, x_{m,u} = 1 \}$ denotes the set of MTs associated with $\mathrm{UAV}_u$. Therefore, the optimal MT assignment rule is expressed as

$$u_m^* = \arg\min_u \frac{2\sqrt{\alpha_{m,u}} \sum_{m' \in \Omega_u} \sqrt{\alpha_{m',u}}}{B_u} \tag{33}$$

For each task $m$, the association variable is derived as

$$x_{m,u}^* = \begin{cases} 1, & u = u_m^\star, \\ 0, & \text{otherwise.} \end{cases}$$

and then, the normalized proportional bandwidth allocation is derived as

$$b^*_{m,u} = \frac{\sqrt{\alpha_{m,u}}}{\sum_{m' \in \Omega_u} \sqrt{\alpha_{m',u}}} B_u. \tag{34}$$

### 4.1.2 Algorithm design

To determine the optimal serving UAV $u^*_m$ for each MT, a delay-aware sequential association procedure is adopted. The key idea is to first prioritize MTs according to their best achievable transmission delay, and then perform UAV association sequentially with adaptive bandwidth reallocation according to Algorithm 1. This weighted cost aware pre-sorting combined with sequential association ensures that MTs with better transmission requirements are prioritized, while preserving the closed-form optimality of bandwidth allocation derived in Subproblem $P_1$ and enabling effective load balancing among UAVs.

The computational complexity of Algorithm 1 is $\mathcal{O}(MU \log M)$, dominated by the MT pre-sorting step and the sequential UAV selection process.

**Algorithm 1** Sequential MT–UAV Association with Closed-Form Bandwidth Allocation

INPUT($\mathcal{M}$,$\mathcal{U}$, $\{B_u\}$, $\{\alpha_{m,u}\}$)
Set of MTs $\mathcal{M}$, set of UAVs $\mathcal{U}$
Available UAV bandwidths $\{B_u\}$
Weighted cost coefficients $\{\alpha_{m,u}\}$

INITIALIZATION
$\Omega_u \leftarrow \emptyset \quad \forall u \in \mathcal{U}$

PRE-SORTING OF MTs (Weighted Cost Ordering)
**for each** $m \in \mathcal{M}$
$u_m^{\min_0} \leftarrow \arg\min_{u \in \mathcal{U}} \left(\sqrt{\alpha_{m,u}}/B_u\right)$
**end for**
Sort MTs in ascending order of $u_m^{\min_0}$
Obtain ordered MT index set $\mathcal{M}_{\text{ord}}$

SEQUENTIAL UAV ASSOCIATION
**for each** $m \in \mathcal{M}_{\text{ord}}$
Select serving UAV $u_m^*$ according to (32)
$\Omega_{u_m^*} \leftarrow \Omega_{u_m^*} \cup \{m\}$
$x^*_{m,u_m} \leftarrow 1$
**end for**

BANDWIDTH ALLOCATION
**for each** $u \in \mathcal{U}$
**for each** $m \in \Omega_u$
Allocate $b^*_{m,u}$ according to (33)
**end for**
**end for**

OUTPUT($\{x^*_{m,u_m}\}, \{b^*_{m,u}\}$)

## 4.2 Subproblem $P_2$: UAV CPU Allocation

For UAV$_u$ with a given set of assigned tasks $\Omega_u$, fixed transmit powers and bandwidth allocated as (34), the CPU computation allocation reduces to a convex optimization subproblem that jointly minimizes processing energy and delay for both MT–UAV and UAV-local computations. The associated Lagrangian term is

$$L_2 = \sum_{m \in \Omega_u} \left(E_m^{\text{cmp}} + \lambda_m \tau_m^{\text{cmp}}\right) + E_u^{\text{cmp}} + \lambda_u \tau_u^{\text{cmp}}. \tag{35}$$

Equivalently, this can be expressed as

$$
\begin{aligned}
P_2: \quad \min_{\substack{f_{u,m},\\ ,f_{u,0}}} \quad & \sum_{m\in\Omega_u} z_{m,u}\Big(\kappa_u C_m D_m f_{u,m}^2 + \lambda_m \frac{C_m D_m}{f_{u,m}}\Big) \\
& + (1-v_u^L)\Big(\kappa_u C_0 D_0 f_{u,0}^2 + \lambda_u \frac{C_0 D_0}{f_{u,0}}\Big) \\
& \text{s.t. (24g)}
\end{aligned} \tag{36}
$$

Building on the dual-decomposition framework, the UAV CPU-allocation model explicitly incorporates the PBFT consensus overhead and frequency reservation required for primary (pr) and non-primary (npr) UAVs. The analytical model computes their required frequencies under the blockchain delay constraint, while allowing overflow computational tasks exceeding this capacity to be offloaded to the LEO tier.

### Closed-Form Optimal CPU Frequencies

Minimizing the computational-energy component in (35) yields closed-form cubic relations between frequency and dual variables. Applying first-order optimality conditions:

$$
\begin{aligned}
& \frac{\partial L_2}{\partial f_{m,u}} = 2\kappa_u C_m D_m f_{m,u} - \lambda_m \frac{C_m D_m}{f_{m,u}^2} = 0, \\
\Rightarrow \quad & f_{m,u}^* = \left(\frac{\lambda_m}{2\kappa_u}\right)^{1/3}, \\
& \frac{\partial L_2}{\partial f_{u,0}} = 2\kappa_u C_u D_u f_{u,0} - \lambda_u \frac{C_u D_u}{f_{u,0}^2} = 0, \\
\Rightarrow \quad & f_{u,0}^* = \left(\frac{\lambda_u}{2\kappa_u}\right)^{1/3}.
\end{aligned} \tag{37}
$$

The obtained optimal frequencies for both UAV internal and MT tasks are combined into a unified frequency pool:

$$\{f_{u,0}^*,\, f_{m,u}^*\}.$$

Sorting these in an ascending order produces a deterministic sequence for sequential allocation:

$$\{f_{(1)} \le f_{(2)} \le \cdots \le f_{(K)}\}.$$

When the cumulative allocated frequency at each UAV exceeds the remaining UAV capacity after PBFT reservation, the corresponding residual tasks (from both UAV internal and MT computations) are offloaded to the LEO for execution. Also, optimization variables $z_{m,u}$ and $v_u^L$ are derived accordingly.

### PBFT Load and Closed-Form Frequencies

Using the computation accounting in (11)–(18), define

$$A = (U+2)(\vartheta_s + \vartheta_c), \qquad B = \vartheta_s + (U-1)\vartheta_c,$$

$$Q = (2\lceil F \rceil + 1)(\vartheta_s + \vartheta_c), \qquad R = \vartheta_s + \vartheta_c.$$

The total workloads are

$$\Lambda_{\mathrm{pr}} = \Lambda_{\mathrm{npr}} = A + 2B + 2Q + R.$$

Although these totals are equal, the primary and non-primary nodes perform different operations at different stages. Let $\Lambda_u = s_u \Lambda_{\mathrm{pr}} + (1 - s_u)\Lambda_{\mathrm{npr}}$ and let $f_u^c$ denote the active consensus frequency at UAV $u$. The primary is selected heuristically as the UAV with the largest CPU capacity, breaking ties by the fewest assigned tasks.

Since the PBFT stages are sequential and each stage maximum is bounded by the corresponding sum,

$$T_b^c \le \widehat{T}_b^c \triangleq \sum_{u=1}^{U} \frac{\Lambda_u}{f_u^c}.$$

We use this conservative bound for reservation, while retaining the sequential stage equations for actual delay evaluation.

Let $A_{\mathrm{blk}} = C_b I_b$ and $\bar{T}_b = \tau_b^{\max} - T_b^p > 0$. Under the present model, other consensus communication delays are neglected. For prescribed reservation limits $0 < \overline{f}_b^L \le F_L$ and $0 < \overline{f}_u^c \le F_u$, solve

$$\begin{aligned} \min_{f_b^L, \{f_u^c\}} \quad & k_L A_{\mathrm{blk}} (f_b^L)^2 + \sum_u \kappa_u \Lambda_u (f_u^c)^2 \\ \text{s.t.} \quad & \frac{A_{\mathrm{blk}}}{f_b^L} + \sum_u \frac{\Lambda_u}{f_u^c} \le \bar{T}_b, \\ & 0 < f_b^L \le \overline{f}_b^L, \qquad 0 < f_u^c \le \overline{f}_u^c. \end{aligned} \tag{38}$$

This conservative convex subproblem is feasible if and only if

$$\frac{A_{\mathrm{blk}}}{\overline{f}_b^L} + \sum_u \frac{\Lambda_u}{\overline{f}_u^c} \le \bar{T}_b. \tag{39}$$

When no upper bound is active, stationarity gives

$$f_b^{L*} = \frac{\xi_b}{k_L^{1/3}}, \qquad f_u^{c*} = \frac{\xi_b}{\kappa_u^{1/3}}, \tag{40}$$

where

$$\xi_b = \frac{A_{\mathrm{blk}} k_L^{1/3} + \sum_u \Lambda_u \kappa_u^{1/3}}{\bar{T}_b}, \qquad \lambda_b = 2\xi_b^3. \tag{41}$$

Here $\lambda_b$ is the blockchain deadline multiplier, distinct from the task-delay multipliers. If the frequencies in (40) violate an upper bound, use

$$f_b^L(\lambda_b) = \min\left\{\overline{f}_b^L, \left(\frac{\lambda_b}{2k_L}\right)^{1/3}\right\},$$

$$f_u^c(\lambda_b) = \min\left\{\overline{f}_u^c, \left(\frac{\lambda_b}{2\kappa_u}\right)^{1/3}\right\}.$$

After verifying (39), find $\lambda_b > 0$ by scalar bisection such that

$$\frac{A_{\text{blk}}}{f_b^L(\lambda_b)} + \sum_u \frac{\Lambda_u}{f_u^c(\lambda_b)} = \bar{T}_b.$$

The resulting frequencies replace those in (40). Independent post-hoc clamping is not used.

The residual UAV task capacity is

$$F_u^{\text{task}} = F_u - f_u^{c*}. \tag{42}$$

$P_2$ allocates only this residual capacity. Tasks that cannot be admitted locally are offloaded when feasible; otherwise the candidate allocation is rejected.

The reservation is optimal only for the conservative blockchain-only subproblem. Its failure does not prove infeasibility under the exact sequential PBFT model. Task allocation remains separate, and the overall method is heuristic.

### 4.3 Subproblem $P_3$: LEO Bandwidth Allocation

After solution of previous subproblems, for offloaded tasks $\ell \in \Psi_L = \{(\text{MT}, m) : y_m^L = 1\} \cup \{(\text{UAV}, u) : v_u^L = 1\}$ (including MT and UAV internal tasks) executed at the LEO, two decoupled relations govern the transmission and computation stages. It is noted that the transmission power associated with overflowed tasks relayed from the same UAV to the LEO satellite is uniformly shared among those tasks. Specifically, if $\text{UAV}_u$ has $|\Psi_L^{(u)}|$ overflowed tasks forwarded to the LEO, the transmit power allocated to each such task satisfies

$$p_\ell = \frac{P_u^L}{|\Psi_L^{(u)}|}, \qquad \forall \ell \in \Psi_L^{(u)}, \tag{43}$$

where $P_u^L$ denotes the total transmit power budget of $\text{UAV}_u$ for LEO offloading.

The LEO-tier bandwidth allocation subproblem aims to minimize the weighted sum of transmission energy and delay cost, subject to the total LEO bandwidth constraint:

$$P_3: \qquad \min_{\{b_\ell\}} \sum_{\ell \in \Psi_L} \left(E_\ell^{\text{tr}} + \lambda_\ell \, \tau_\ell^{\text{tr}}\right)$$

$$\text{s.t.} \quad \sum_{\ell \in \Psi_L} b_\ell \le B_L,$$
$$\tau_\ell^{\text{tr}} = \frac{D_\ell}{b_\ell \log_2\left(1 + \frac{h_\ell p_\ell}{\sigma^2}\right)}, \qquad E_\ell^{\text{tr}} = p_\ell \tau_\ell^{\text{tr}}$$

The corresponding Lagrangian function (with dual variable $\gamma_L \ge 0$ for the bandwidth constraint) is expressed as

$$\begin{aligned} L_3 &= \sum_{\ell \in \Psi_L} \left(p_\ell \tau_\ell^{\text{tr}} + \lambda_\ell \tau_\ell^{\text{tr}}\right) + \gamma_L \left(\sum_{\ell \in \Psi_L} b_\ell - B_L\right) \\ &= \sum_{\ell \in \Psi_L} \left(\frac{(p_\ell + \lambda_\ell) D_\ell}{b_\ell \log_2\left(1 + \frac{h_\ell p_\ell}{\sigma^2}\right)} + \gamma_L b_\ell\right) - \gamma_L B_L \end{aligned} \tag{44}$$

Defining the condensed term

$$\beta_{L,\ell} \triangleq \frac{(p_\ell + \lambda_\ell) D_\ell}{\log_2\left(1 + \frac{h_\ell p_\ell}{\sigma^2}\right)} \tag{45}$$

and taking derivative, yields

$$\frac{\partial L_3}{\partial b_\ell} = -\frac{\beta_{L,\ell}}{b_\ell^2} + \gamma_L = 0, \quad \Rightarrow \quad b_\ell = \sqrt{\frac{\beta_{L,\ell}}{\gamma_L}}. \tag{46}$$

Using the equality constraint $\sum_\ell b_\ell = B_L$, one obtains

$$\sum_{\ell \in \Psi_L} \sqrt{\frac{\beta_{L,\ell}}{\gamma_L}} = B_L \quad \Rightarrow \quad \sqrt{\gamma_L} = \frac{\sum_{\ell \in \Psi_L} \sqrt{\beta_{L,\ell}}}{B_L} \tag{47}$$

Finally, the closed-form optimal bandwidth allocation is given by

$$b_\ell^* = \frac{\sqrt{\beta_{L,\ell}}}{\sum_{\ell \in \Psi_L} \sqrt{\beta_{L,\ell}}} B_L. \tag{48}$$

This expression follows the proportional-allocation (water-filling-like) structure, ensuring minimal weighted latency–energy cost under the total LEO bandwidth constraint.

## 4.4 Subproblem $P_4$: LEO Task CPU Allocation

The blockchain frequency $f_b^{L*}$ is fixed by (38) and is not reoptimized during task allocation. The residual LEO CPU constraint is

$$\sum_{\ell \in \Psi_L} f_\ell \le F_L - f_b^{L*} \triangleq F_L^\star. \tag{49}$$

For a nonempty offloaded-task set, $F_L^\star > 0$ is required. Otherwise, the candidate is rejected.

### LEO Computation Optimization

Given the reduced budget (49), the remaining optimization minimizes the weighted sum of computation energy and delay for all offloaded tasks:

$$\begin{aligned} P_4: \quad & \min_{\{f_\ell\}} \sum_{\ell \in \Psi_L} \left(E_\ell^{\mathrm{cmp}} + \lambda_\ell \tau_\ell^{\mathrm{cmp}}\right) \\ & \text{s.t.} \quad \tau_\ell^{\mathrm{cmp}} = \frac{C_\ell D_\ell}{f_\ell}, \qquad E_\ell^{\mathrm{cmp}} = \kappa_L C_\ell D_\ell f_\ell^2. \end{aligned} \tag{50}$$

To enforce the total LEO CPU constraint, we introduce the Lagrange multiplier $\mu_L$. The Lagrangian becomes

$$L_4 = \sum_{\ell \in \Psi_L} \left(\kappa_L C_\ell D_\ell f_\ell^2 + \lambda_\ell \frac{C_\ell D_\ell}{f_\ell}\right) + \mu_L \left(\sum_{\ell \in \Psi_L} f_\ell - F_L^\star\right).$$

The KKT stationarity condition yields

$$\frac{\partial L_4}{\partial f_\ell} = 2\kappa_L C_\ell D_\ell f_\ell - \lambda_\ell \frac{C_\ell D_\ell}{f_\ell^2} + \mu_L = 0. \tag{51}$$

Multiplying both sides by $f_\ell^2$ gives the cubic equation

$$2\kappa_L C_\ell D_\ell f_\ell^3 + \mu_L f_\ell^2 - \lambda_\ell C_\ell D_\ell = 0, \tag{52}$$

which yields a unique real positive root for each task.

**Solving for $\mu_L$.** For positive task-delay multipliers, first compute

$$f_\ell(0) = \left(\frac{\lambda_\ell}{2\kappa_L}\right)^{1/3}.$$

If $\sum_{\ell \in \Psi_L} f_\ell(0) \le F_L^\star$, set $\mu_L^\star = 0$. Otherwise, use bisection to find $\mu_L^\star > 0$ satisfying

$$\sum_{\ell \in \Psi_L} f_\ell(\mu_L^\star) = F_L^\star,$$

where each frequency is the unique positive root of (52). If $\Psi_L$ is empty, no task-frequency allocation is needed.

## 4.5 Delay-Multiplier Update

Task-delay multipliers are updated using the delay residuals of the sequential heuristic:

$$\begin{aligned}
\lambda_m^{(t+1)} &= \Big[\lambda_m^{(t)} + \delta_t\big(\tau_m^{\mathrm{tot}} - \tau_m^{\max}\big)\Big]_{\lambda_{\min}}, \\
\lambda_u^{(t+1)} &= \Big[\lambda_u^{(t)} + \delta_t\big(\tau_u^{\mathrm{tot}} - \tau_u^{\max}\big)\Big]_{\lambda_{\min}}, \\
\delta_t &= \frac{\delta_0}{\sqrt{t}}, \qquad t \geq 1,
\end{aligned} \tag{53}$$

where $\delta_t$ denotes the adaptive diminishing step size at iteration $t$ and $[z]_{\lambda_{\min}} = \max\{\lambda_{\min}, z\}$ and $\lambda_{\min} > 0$ is a small numerical floor that prevents zero task-frequency candidates. These updates are used within a sequential heuristic; global optimality and convergence to a stationary point of the original MINLP are not guaranteed.

## 4.6 Sequential Dual Decomposition Algorithm (SDD)

The delay-only dualized problem (P) is decomposed into four tractable subproblems: $P_1$ (MT–UAV bandwidth allocation and assignment), $P_2$ (UAV CPU frequency optimization), $P_3$ (LEO bandwidth allocation), and $P_4$ (LEO task CPU allocation). The overall solution is obtained via a sequential dual decomposition (SDD) algorithm with iterative refinement of MT–UAV association in Algorithm 2.

At each iteration, the association heuristic and continuous-resource updates are applied sequentially for fixed dual variables, and the resulting delay violations are used to update the Lagrange multipliers via a subgradient method. The iterative refinement phase is an integral component of the proposed algorithm and is executed at every dual iteration. To avoid oscillatory associations, MT reassignment is permitted only when a strictly lower dual-aware association cost is achieved, controlled by a stabilization threshold $\varepsilon_{\mathrm{switch}}$.

**Algorithm 2** Sequential Dual Decomposition (SDD)

**Input:** System parameters, channel states, task profiles
**Output:** MT–UAV association, resource allocation, dual variables
Initialize $\boldsymbol{\lambda} \leftarrow \lambda_0 \mathbf{1}$, where $\lambda_0 \geq \lambda_{\min} > 0$
Set maximum iteration count $t_{\max}$
Set iteration index $t \leftarrow 0$
**repeat**
  $t \leftarrow t + 1$
  **Dual-aware cost update:**
  **for** each $m \in \mathcal{M}$ and $u \in \mathcal{U}$ **do**
    Update weighted cost coefficient $\alpha_{m,u}(\boldsymbol{\lambda})$ in (28)
  **end for**
  **Inner Sequential Association:**
  Run Algorithm 1 with updated $\{\alpha_{m,u}\}$
  **Iterative Refinement with Stabilized Switching:**
  **for** each $m \in \mathcal{M}$ **do**
    **if** $\mathcal{C}_{m,u_{\text{new}}} < (1 - \varepsilon_{\text{switch}})\mathcal{C}_{m,u_{\text{old}}}$ **then**
      Reassign $\text{MT}_m$
    **end if**
  **end for**
  Obtain association $\{x^*_{m,u}\}$ and bandwidth $\{b^*_{m,u}\}$
  **Primal Resource Allocation:**
  Recompute bandwidths for the refined association
  Check (39); if it fails, report conservative reservation failure and stop
  Compute the blockchain reservation
  Solve $P_2$, $P_3$, and $P_4$ sequentially using residual CPU capacities and bisection where needed
  Compute $\tau_m^{\text{tr}}, \tau_m^{\text{cmp}}, \tau_u^{\text{tr}}, \tau_u^{\text{cmp}}$
  **Dual Update:**
  Update $\boldsymbol{\lambda}$ using (53)
**until** $\max_m |\tau_m^{\text{tot}} - \tau_m^{\max}| \leq \varepsilon$ **and** $\max_u |\tau_u^{\text{tot}} - \tau_u^{\max}| \leq \varepsilon$.

# 5 Performance Evaluation

In this section, we evaluate the performance of the proposed sequential dualization framework for joint task offloading and resource allocation in the MT–UAV–LEO architecture with blockchain support. All simulations are conducted using MATLAB with parameter values in Table 2 and results are averaged over multiple independent Monte Carlo (MC) realizations. Unless otherwise stated, MTs and UAVs are randomly deployed within the service area in each realization like Fig. 2. Task sizes, computation densities, and channel conditions follow the system model described in Section III. The maximum tolerable delay constraints are enforced for MT tasks, UAV internal tasks, and blockchain block generation. For fair comparison, only MC realizations in which

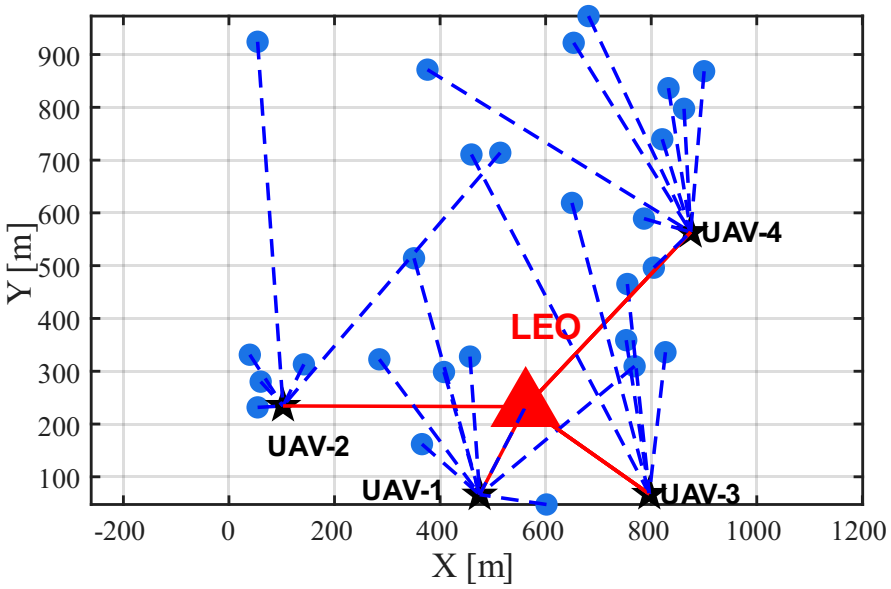



**Fig. 2**: One network topology realization

all compared schemes satisfy the feasibility conditions are included when computing average energy consumption.

## 5.1 Benchmark Description

This work compares several benchmark schemes, each representing a different combination of MT–UAV association, radio bandwidth allocation, UAV CPU scheduling, and LEO-tier resource management. The benchmarks are designed to highlight the impact of association optimality, bandwidth-aware allocation, and geometric programming (GP) versus Lagrangian-based resource optimization.

- **Non-iterative Sequential Association (NSA)**: A non-iterative benchmark corresponding to the special case $\lambda_m = 0$ of the proposed sequential assignment framework. MT–UAV association is obtained via sequential assignment considering only the UAV bandwidth constraint. After association and bandwidth allocation, UAV CPU frequencies are computed from the residual delay budget following [7]. Tasks are then sorted by computational demand, and those exceeding UAV computational capacity are forwarded to the LEO layer. In the second layer, bandwidth is again reallocated under the sum-bandwidth constraint, and CPU frequency is again determined from the remaining delay budget.
- **Maximum Channel Gain-GP (MCG-GP)**: MT–UAV association is performed greedily based on maximum channel gain. Given the association, MT→UAV bandwidth and UAV CPU frequency are optimized via geometric programming (GP) following [3]. UAV CPU capacity constraints are not enforced during association;

**Table 2**: Simulation Parameters (Complete Set)

| Parameter | Value |
|---|---|
| Number of MTs ($M$) | 8-32 |
| Number of UAVs ($U$) | 4 |
| UAV altitudes ($H_{\text{UAV}}$) | [100, 150, 200, 250] m |
| LEO altitude ($H_{\text{LEO}}$) | 400 km |
| MT data size ($D_m$) | $\sim \mathcal{U}[8, 10]$ Mbits |
| Mbits UAV data size ($D_u$) | 8 Mbits |
| CPU cycles/bit ($C_m, C_u$) | $\sim \mathcal{U}[100, 1000]$ |
| UAV CPU frequencies ($F_u$) | $\sim \mathcal{U}[1, 10]$ GHz |
| LEO CPU frequency ($F_L$) | $\sim \mathcal{U}[10, 100]$ GHz |
| MT deadline ($\tau_m^{\max}$) | 5 s |
| UAV deadline ($\tau_u^{\max}$) | 5 s |
| $\varepsilon_{\text{switch}}$ | 1e-2 |
| MT→UAV bandwidth ($B_u$) | 10 MHz / UAV |
| UAV→LEO bandwidth ($B_L$) | 15 MHz |
| Noise power MT→UAV ($\sigma_s^2$) | $17 \times 10^{-12}$ [2] |
| Noise power UAV→LEO ($\sigma_u^2$) | $10 \times 10^{-15}$ [2] |
| Max MT/UAV transmit power ($p_{\max}$) | 2 / 10 W |
| UAV CPU coefficient ($k_U$) | $1.2 \times 10^{-27}$ [2] |
| LEO CPU coefficient ($k_L$) | $1.2 \times 10^{-27}$ [2] |
| Block size ($I_b$) | $2 \times 10^6$ bits |
| Block intensity ($C_b$) | $5 \times 10^3$ cycles/bit |
| Blockchain completion deadline ($\tau_b^{\max}$) | 5 s |
| PBFT stages ($\theta_s$, $\theta_c$) | $10^4$, $10^4$ |
| Carrier frequency ($f_c$) | 2/12 GHz UAV/LEO |
| Speed of light ($c_0$) | $3 \times 10^8$ m/s |
| UAV path-loss ($a_1, a_2$) | 9.61, 0.16 [2] |
| LoS/NLoS losses ($N_{\text{LoS}}, N_{\text{NLoS}}$) | 1, 20 dB [2] |
| LEO→UAV distance ($d_{\text{LEO}-\text{UAV}}$) | 400 km |
| Antenna gains ($G_t^{UAV}, G_r^{LEO}$) | 12 dB, 44dB UAV-LEO |
| UAV-LEO $\Delta_0$ | 5dB |
| MT coordinates | $\sim \mathcal{U}(0, 1000)$ m |
| UAV coordinates | $\sim \mathcal{U}(0, 1000)$ m |

overflowed tasks are forwarded to the LEO layer, where GP is re-applied for bandwidth and CPU allocation. This benchmark represents channel-driven association without bandwidth-aware assignment.

- **Uniform Association-GP (Uni-GP)**: Identical to MCG-GP except that MT–UAV association is uniformly random, with an equal number of MTs assigned to each UAV.
- **Optimal Association-GP (Opt-GP)**: MT–UAV association is optimized considering only the UAV bandwidth constraint. Given the optimized association, MT→UAV bandwidth and UAV CPU frequencies are jointly optimized via GP, similar to MCG-GP and Uni-GP. This benchmark represents the best possible association under GP-based resource allocation.

## 5.2 Monte-Carlo Setup

Monte-Carlo simulations are performed over $N_{\text{MC}} = 100$ independent network realizations. In each realization, MT and UAV locations, task parameters, and channel

conditions are randomly generated. For each MC iteration, aggregate all energy and delay statistics. For fair comparison, average energy values are computed only over realizations in which all compared schemes satisfy the feasibility constraints.

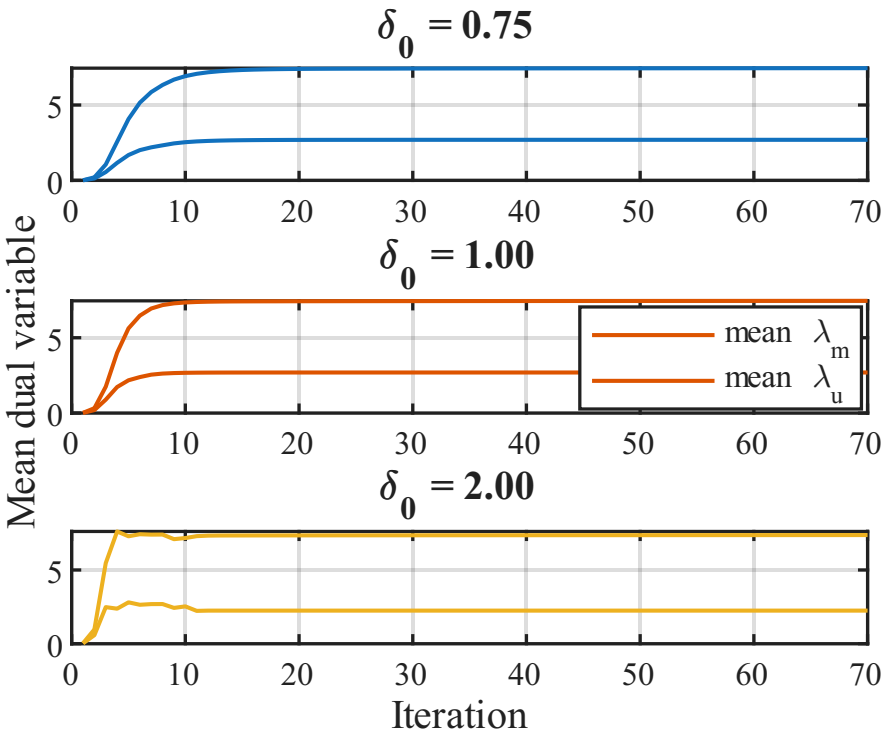



**Fig. 3**: Convergence of the average dual variables in the SDD method

### 5.3 Convergence of the Proposed SDD Algorithm

Fig. 3 illustrates the convergence behavior of the proposed SDD algorithm for 20 MTs and $\delta 0 = [.75, 1, 2]$ as the initial step size values. The mean of dual variables converges within a limited number of iterations, typically fewer than 20, indicating rapid empirical stabilization in the tested scenarios and stability of the Sequential Dualization framework. The small variance across MC realizations further highlights the robustness of the proposed method.

### 5.4 Average Energy Consumption vs. Number of MTs

Fig. 4 depicts the average total system energy consumption as a function of the number of MTs. As the number of MTs increases, the overall energy consumption rises for all schemes due to higher communication and computation loads. The proposed SDD scheme consistently achieves the lowest energy consumption across all MT densities.

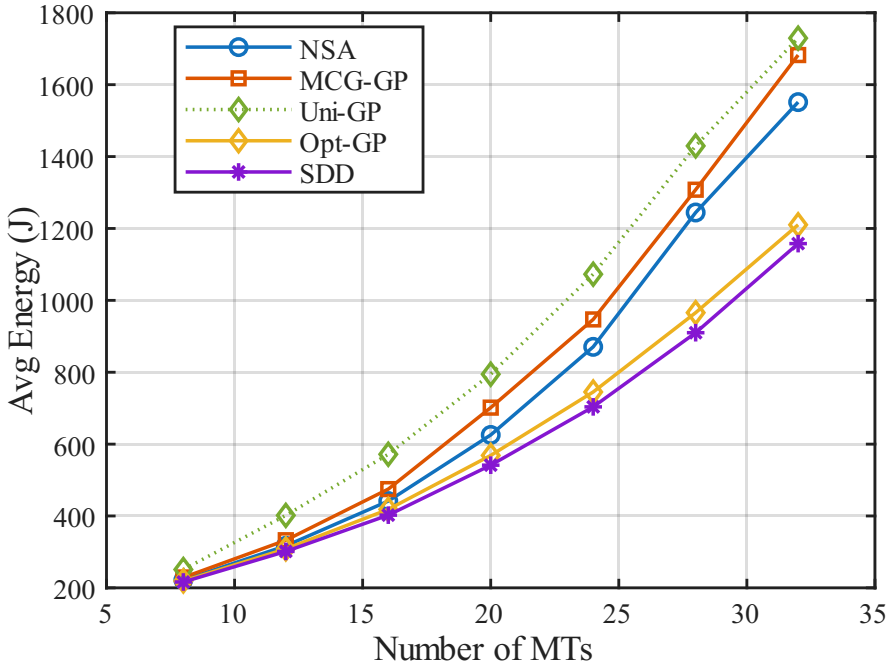



**Fig. 4**: Energy vs. number of MT devices .

In contrast, MCG-GP and Uni-GP exhibits significantly higher energy consumption, mainly due to unbalanced MT–UAV association and excessive task relaying to the LEO layer.

## 5.5 Outage Probability Performance vs. Number of MTs

Fig. 5 illustrates the outage probability versus the number of MTs, where outage is defined as the fraction of MC realizations violating at least one delay constraint. NSA shows slightly higher outage due to the lack of dual-variable refinement. MCG and Uniform suffer from rapidly increasing outage as the network densifies, indicating poor feasibility. SDD achieves the best performance.

### 5.5.1 Impact of Delay Tolerance

Fig. 6(a) and 7(a) illustrates the average energy consumption and feasibility for 28 MTs versus the maximum tolerable delay. When the delay constraint is tight, only energy-intensive realizations remain feasible, resulting in higher average energy. As the delay tolerance increases, more realizations become feasible and the required energy decreases accordingly.

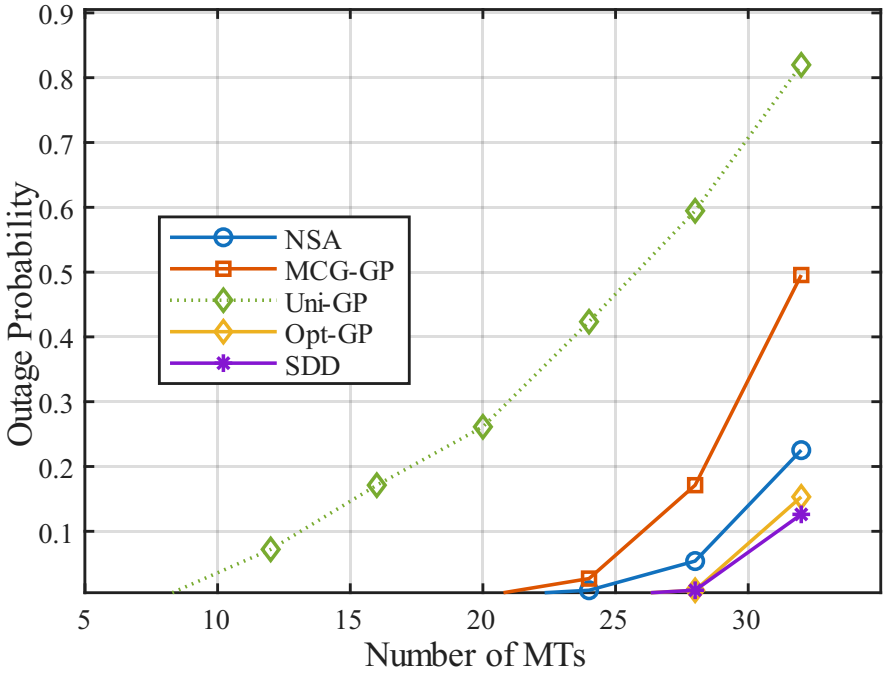



**Fig. 5**: Outage vs. number of MT devices .

### 5.5.2 Impact of UAV Bandwidth

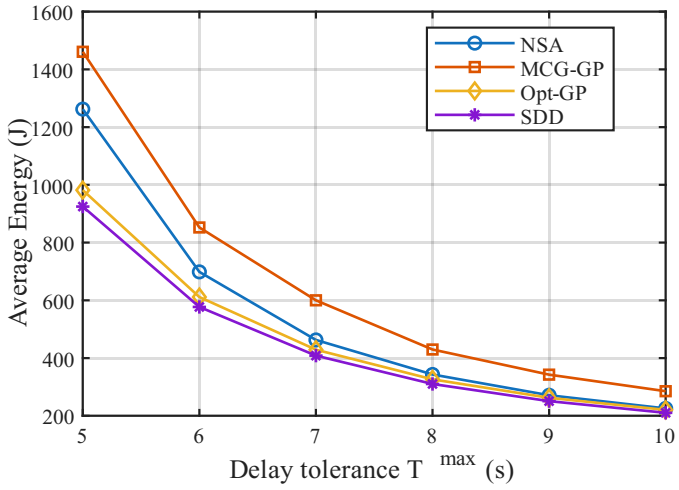
Fig. 6(b) and 7(b) shows the effect of the maximum UAV bandwidth on system energy consumption. Larger UAV bandwidth significantly improves energy efficiency by increasing MT–UAV transmission rates and reducing delay pressure. The proposed SDD and NSA schemes efficiently exploit the available bandwidth, whereas MCG fails to balance bandwidth among UAVs.

### 5.5.3 Impact of LEO Computing Capacity

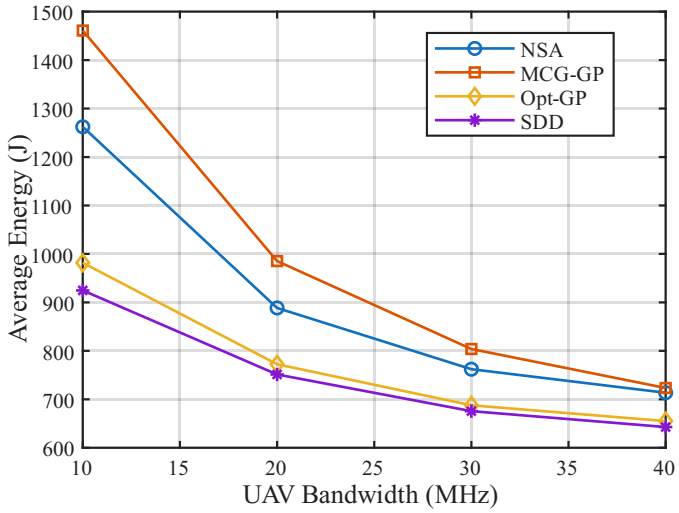
Fig. 6(c) and 7(c) presents the average energy and feasibility consumption versus the maximum LEO CPU frequency. As the feasibility of high demanding cases improves due to better CPU budget, energy increases respectively. While all methods benefit from larger CPU, the MCG and NSA feasibility reach a fixed level which indicate they are limited by bandwidth rather than CPU.

## 6 Conclusion

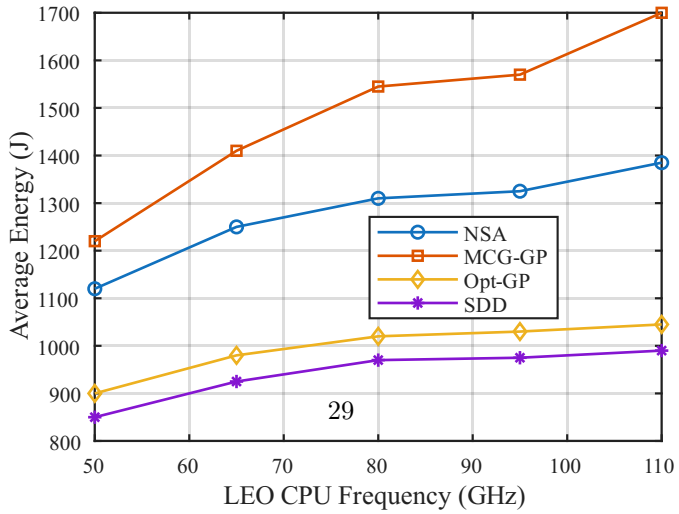
According to the simulation results, the proposed iterative Lagrangian-based methods for joint association and resource allocation in the LEO–UAV–MT network consistently outperform all benchmark schemes. The proposed *SDD* method achieves the lowest total energy consumption and the highest feasibility rate compared with GP-based approaches under three association strategies, namely maximum channel gain,




(a) Delay tolerance $T_i^{\max}$

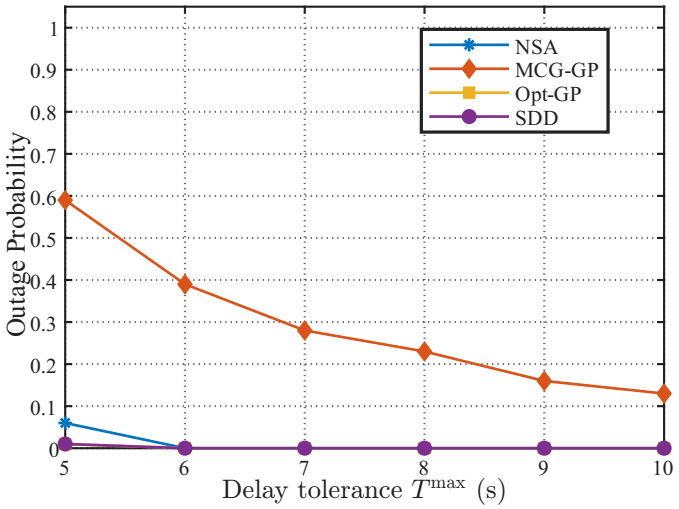



(b) UAV bandwidth $B_u^{\max}$

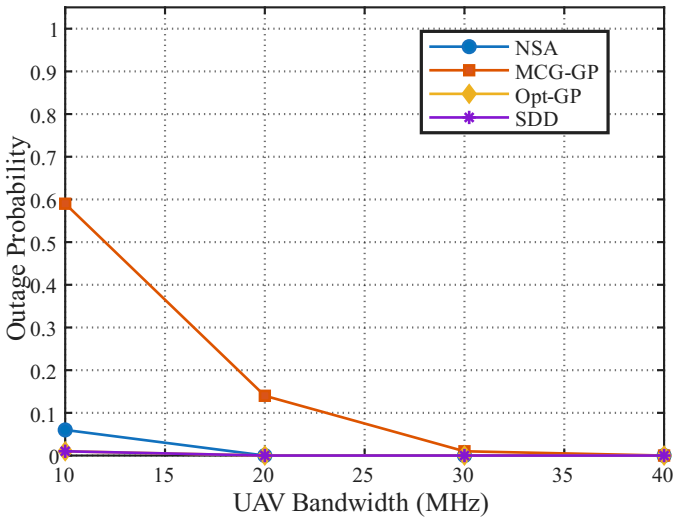



(c) LEO computing capacity $F_L^{\max}$

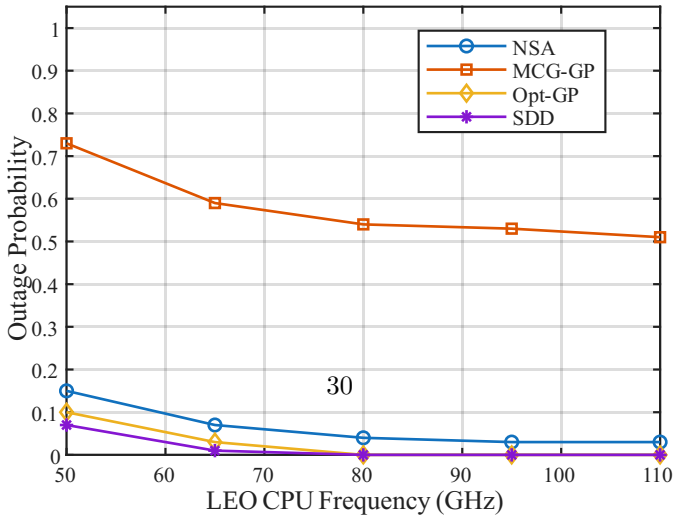
**Fig. 6**: Average energy consumption versus key system parameters




(a) Delay tolerance $T_i^{\max}$




(b) UAV bandwidth $B_u^{\max}$




(c) LEO computing capacity $F_L^{\max}$

**Fig. 7**: Outage versus key system parameters

uniform, and bandwidth-aware association. Moreover, the NSA method, despite its significantly lower computational complexity, demonstrates better energy performance than the MCG-GP and MCG-Uniform methods. In contrast, the MCG and uniform association schemes suffer from severe load imbalance among UAVs, which leads to inefficient resource utilization and degraded overall energy performance. The extension to optimized transmit power is left for future work.

## Data availability

The simulation parameters used and analysed during the current study are included in this published article. The datasets generated during the current study are available from the corresponding author on reasonable request.

## References


[1] J. Chen, Z. Kuang, Y. Zhang, S. Lin, and A. Liu, "Blockchain-enabled computing offloading and resource allocation in multi-UAVs MEC network: A Stackelberg game learning approach," *IEEE Trans. Inf. Forensics Security*, vol. 20, pp. 3632–3645, 2025.

[2] J. Du, J. Wang, A. Sun, J. Qu, J. Zhang, C. Wu, and D. Niyato, "Joint optimization in blockchain- and MEC-enabled space–air–ground integrated networks," *IEEE Internet Things J.*, vol. 11, no. 19, pp. 31862–31877, 2024.

[3] M. Fathi, M. Saroughi, and A. Zareie, "Delay-sensitive task offloading optimization by geometric programming," *IEEE Trans. Cloud Comput.*, vol. 12, no. 3, pp. 889–896, 2024.

[4] J. Li, Y. Shi, C. Dai, C. Yi, Y. Yang, X. Zhai, and K. Zhu, "A learning-based stochastic game for energy efficient optimization of UAV trajectory and task offloading in space/aerial edge computing," *IEEE Trans. Veh. Technol.*, vol. 74, no. 6, pp. 9717–9733, 2025.

[5] M. D. Nguyen, L. B. Le, and A. Girard, "Computation offloading, UAV placement, and resource allocation in SAGIN," in *Proc. IEEE GLOBECOM Workshops*, 2022, pp. 1413–1418.

[6] Q. Tang, Z. Fei, B. Li, and Z. Han, "Computation offloading in LEO satellite networks with hybrid cloud and edge computing," *IEEE Internet Things J.*, vol. 8, no. 11, pp. 9164–9176, 2021.

[7] M. Valitabar, M. Fathi, and K. Navaie, "Efficient resource allocation for blockchain-enabled mobile edge computing: A joint optimization approach," *IEEE Access*, vol. 13, pp. 129011–129023, 2025.

[8] S. Xi, B. Shang, H. Zhang, J. Ma, and P. Fan, "Energy optimization in multisatellite-enabled edge computing systems," *IEEE Internet Things J.*, vol. 11,

no. 12, pp. 21715–21726, 2024.

[9] L. Yu, B. Li, Y. Yao, Z. Wang, Z. Li, Z. Dong, and D. Cai, "Efficient and emission-reducing blockchain-enabled multi-UAV-assisted MEC system in IoT networks," *IEEE Internet Things J.*, vol. 11, no. 24, pp. 40645–40655, 2024.

[10] H. Zhang, S. Xi, H. Jiang, Q. Shen, B. Shang, and J. Wang, "Resource allocation and offloading strategy for UAV-assisted LEO satellite edge computing," *Drones*, vol. 7, no. 6, 2023.

[11] Z. Zhang, K. Zeng, and Y. Yi, "Blockchain-empowered secure aerial edge computing for AIoT devices," *IEEE Internet Things J.*, vol. 11, no. 1, pp. 84–94, 2024.

[12] C. Zhao, M. Li, P. Si, R. Yang, Z. Wang, and Y. Zhang, "Resource allocation and optimization for UAV-assisted IoT based on MEC and blockchain," in *Proc. Int. Conf. Comput. Commun. (ICCC)*, 2021, pp. 2080–2086.

[13] J. Feng, F. R. Yu, Q. Pei, J. Du, and L. Zhu, "Joint optimization of radio and computational resources allocation in blockchain-enabled mobile edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 19, no. 6, pp. 4321–4334, 2020.

[14] X. Hu, K.-K. Wong, K. Yang, and Z. Zheng, "UAV-assisted relaying and edge computing: Scheduling and trajectory optimization," *IEEE Trans. Wireless Commun.*, vol. 18, no. 10, pp. 4738–4752, 2019.

[15] Z. Yu, Y. Gong, S. Gong, and Y. Guo, "Joint task offloading and resource allocation in UAV-enabled mobile edge computing," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3147–3159, 2020.

[16] Q. Tang, Z. Fei, J. Zheng, B. Li, L. Guo, and J. Wang, "Secure aerial computing: Convergence of mobile edge computing and blockchain for UAV networks," *IEEE Trans. Veh. Technol.*, vol. 71, no. 11, pp. 12073–12087, 2022.

[17] Z. Yang, C. Pan, K. Wang, and M. Shikh-Bahaei, "Energy efficient resource allocation in UAV-enabled mobile edge computing networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 9, pp. 4576–4589, 2019.

[18] Y. K. Tun, T. N. Dang, K. Kim, M. Alsenwi, W. Saad, and C. S. Hong, "Collaboration in the sky: A distributed framework for task offloading and resource allocation in multi-access edge computing," *IEEE Internet Things J.*, vol. 9, no. 23, pp. 24221–24235, 2022.