

# Joint UAV Trajectory Planning and LEO Satellite Selection for Data Offloading in Space-Air-Ground Integrated Networks

Tie Liu <sup>1</sup> , Firstname Lastname <sup>2</sup> and Firstname Lastname <sup>2,\*</sup>

<sup>1</sup> Affiliation 1; e-mail@e-mail.com

<sup>2</sup> Affiliation 2; e-mail@e-mail.com

\* Correspondence: e-mail@e-mail.com; Tel.: (optional; include country code; if there are multiple corresponding authors, add author initials) +xx-xxxx-xxx-xxxx (F.L.)

## Abstract

With the development of low earth orbit (LEO) satellites and unmanned aerial vehicles (UAVs), the space-airground integrated network (SAGIN) becomes a major trend in the next-generation networks. However, due to the instability of heterogeneous communication and time-varying characteristics of SAGIN, it is challenging to meet the remote Internet of Things (IoT) demands for data collection and offloading. In this paper, we investigate a two-phase hierarchical data uplink model in SAGIN. Specifically, UAVs optimize trajectories to enable efficient data collection from IoT devices, and then they transmit the data to LEO satellites with computing capabilities for further processing. The problem is formulated to minimize the total energy consumption for UAVs and the packet loss rate transmitted to LEO satellites. Ultimately, simulation results demonstrate the effectiveness of the proposed algorithm, which reduces energy consumption by approximately 10% compared to the baseline algorithm and lowers the packet loss rate by an average of 3 to 5 times.

**Keywords:** SAGIN, UAV, LEO satellite, data offloading.

## 1. Introduction

The Internet of Things (IoT) devices are widely applied in the daily life, such as environmental monitoring and traffic management. However, due to the limited ground base stations in remote or post-disaster areas, it is difficult to satisfy the demands for data collection and offloading supported by the terrestrial networks. The space-air-ground integrated network (SAGIN) is perceived as an effective solution to tackle the above difficulties [1]. In SAGIN, low earth orbit (LEO) satellites can provide the IoT devices with extensive connectivities [2] [3]. Additionally, the in-orbit computing allows LEO satellites to directly process tasks, which avoids the long propagation delays and eases the congestion on bandwidth-limited downlink channels [4] [5]. Moreover, unmanned aerial vehicles (UAVs), as ideal candidates for aerial relays, can be deployed flexibly to ensure efficient data collection [6]. On one hand, the UAVs trajectories can be optimized to minimize the multi-hop transmission and propagation distance [7]. Besides, UAVs facilitate the line-of-sight (LoS) communications with ground devices for a wide view, improving the channel quality and enhancing the transmission throughput [8] [9]. Nevertheless, the limitation of communication resources restricts the number of IoT devices served by UAVs and leads to a poor spectrum efficiency [10]. In response to this issue, the non-orthogonal multiple access (NOMA) technology, which emerges as a promising paradigm, allows multiple IoT devices to share a single resource block.

Received:

Revised:

Accepted:

Published:

**Citation:** L, Tie.; Lastname, F.; Lastname, F. Joint UAV Trajectory Planning and LEO Satellite Selection for Data Offloading in Space-Air-Ground Integrated Networks. *Journal Not Specified* **2025**, *1*, 0. <https://doi.org/>

**Copyright:** © 2025 by the authors. Submitted to *Journal Not Specified* for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Some works have begun to explore problems on resource allocations in SAGIN. The authors in [11] propose an iterative power allocation algorithm to maximize the sum rate in a NOMA-based hybrid satellite-UAV-terrestrial network. In [12], the authors consider the complexity of SAGIN and solve the service function chain scheduling problem by incorporating deep reinforcement learning. The authors in [13] study the total energy consumption minimization for task processing in an SAGIN-supported mobile edge computing system. In [14], the authors introduce a data collection scheme to balance the throughput and fairness among the IoT nodes in SAGIN. Although the above works are conducted in SAGIN, the satellite selection issues are not considered, which can significantly enhance the performance of the system.

Given the instability and time-varying nature of heterogeneous communications within the SAGIN system, we propose a hierarchical framework integrating ground-based IoT devices, unmanned aerial vehicles (UAVs) serving as aerial relays, and computationally capable low-Earth orbit satellites. The problem is modelled as minimising total UAV energy consumption whilst ensuring service quality guarantees. To address this, the process is divided into two phases. In Phase One, we design algorithms for IoT pairing, power allocation, and UAV trajectory planning. In the second phase, we introduce a demand-aware flexible switching mechanism for low-Earth orbit satellites.

The remainder of this paper is organized as follows. In Section II, we design the system model and provide the problem formulation. In Section III, the algorithms are proposed. Section IV evaluates the performance of the proposed algorithms via numerical analyses. Finally, the conclusions are drawn in Section V.

## 2. Materials and Methods

As shown in Fig. 1, we consider an SAGIN which consists of  $U$  UAVs denoted by  $\mathcal{U} = \{1, 2, \dots, U\}$ , and  $S$  LEO satellites indicated by  $\mathcal{S} = \{1, 2, \dots, S\}$ . In addition,  $K$  IoT devices scattered randomly on the ground are represented as  $\mathcal{K} = \{1, 2, \dots, K\}$ . Due to the limited computational capabilities of IoT terminals, the data they collect must be uploaded to low-Earth orbit satellites for further processing. However, constrained by their own energy and transmission power limitations, IoT devices struggle to communicate directly with these satellites. To address this, we introduce drones as aerial base stations to assist in data aggregation and forwarding within target areas. Based on this architecture, the data upload process can be divided into two stages. In the first stage, the UAV follows a pre-planned flight path to sequentially reach multiple hovering positions, collecting data generated by IoT devices within the area before returning to the starting point upon completion. In the second stage, after returning to the starting point, the UAV hovers at the origin and selects an appropriate LEO satellite to perform computational offloading tasks.

### 2.1. Data Collection from IoT to UAV

During the data upload phase for IoT devices and drones, we introduce Non-Orthogonal Multiple Access (NOMA) technology to enhance the system's spectrum efficiency and transmission effectiveness. Specifically, we assume that within a defined spatial range, any two IoT devices can form a cooperative pair, transmitting their uplink data simultaneously to the UAV via the NOMA mechanism. For devices that cannot meet pairing conditions or fail to pair successfully, Orthogonal Frequency Division Multiple Access (OFDMA) is employed for independent data transmission.

A three-dimensional Cartesian coordinate system is employed to characterize the spatial locations of UAVs and IoT devices. The IoT devices are assumed to be randomly distributed on the ground plane, and the horizontal coordinate of the  $k$ -th device denoted as  $\mathbf{q}_k = (x_k, y_k, 0)$ . Each UAV operates at a fixed altitude  $h_u$ . Accordingly, the position of

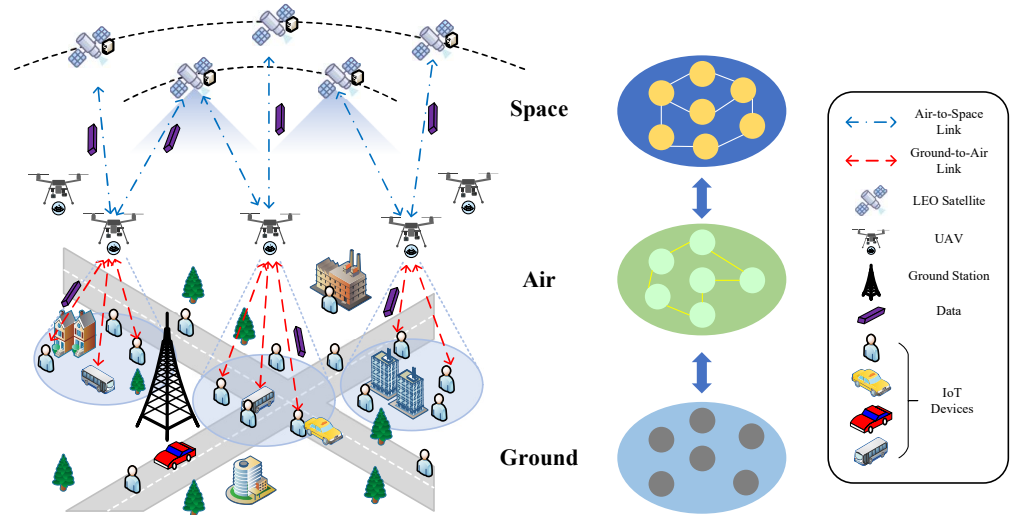


Figure 1. figure1.

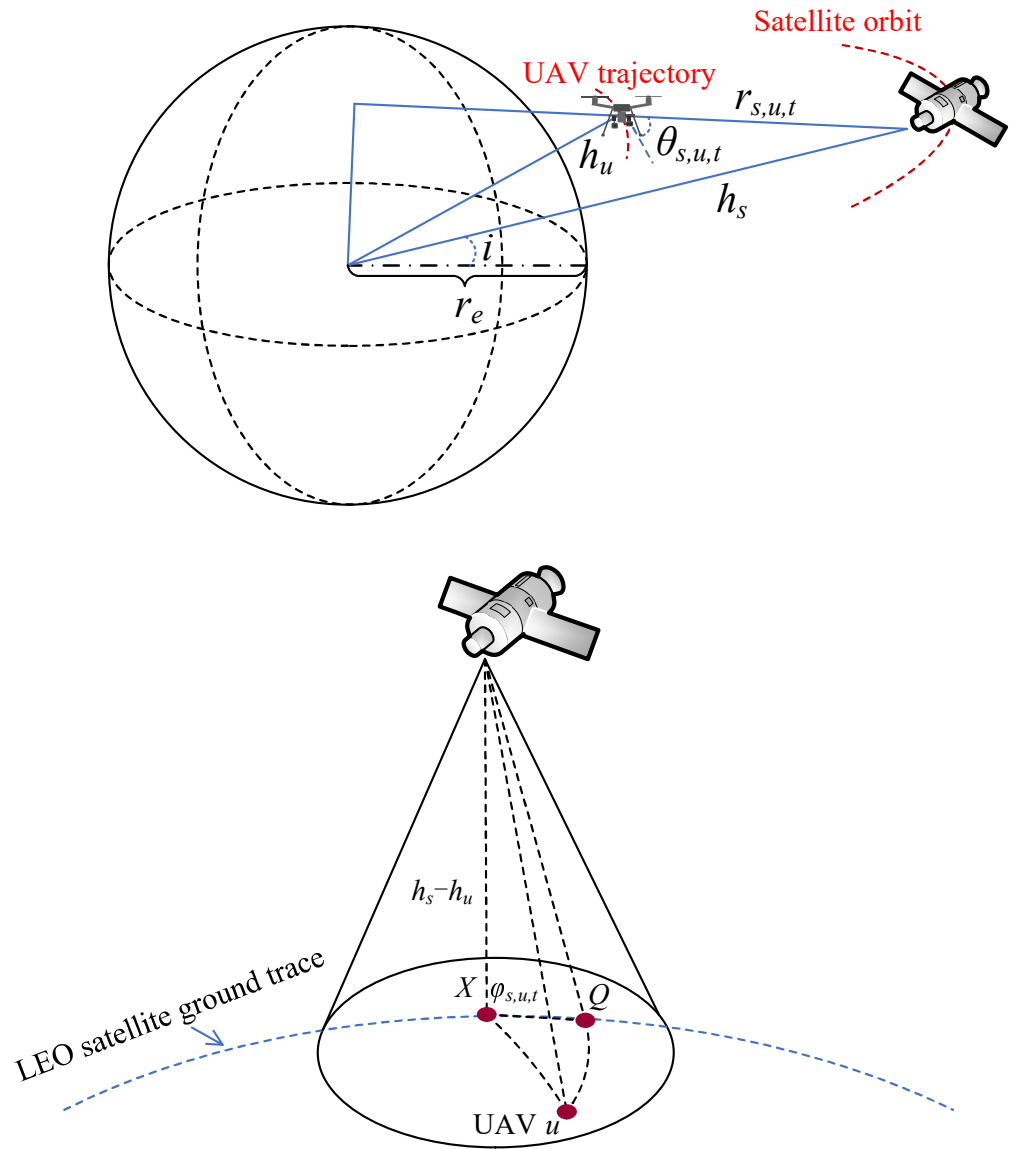


Figure 2. figure2.

the  $u$ -th UAV at the  $n$ -th hover point is expressed as  $\mathbf{q}_u(n) = (x_u(n), y_u(n), h_u)$ . Due to the elevated UAV altitude and the unobstructed propagation environment, the wireless links

between IoT devices and UAVs are dominated by line-of-sight (LoS) propagation. Under this condition, the channel gain between IoT device  $k$  and its associated UAV is modeled as

$$G_k = \frac{\beta_0}{h_u^2 + (x_u(n) - x_k)^2 + (y_u(n) - y_k)^2}, \quad (1)$$

where  $\beta_0$  represents the channel gain at the reference distance  $r_0 = 1\text{m}$ .

Let  $\mathcal{P}$  denote the set of all feasible transmission pairs, where each pair  $p \in \mathcal{P}$  consists of either (i) two IoT devices forming a NOMA cluster or (ii) a single device operating in OFDMA mode. A binary decision variable  $\delta_p \in \{0, 1\}$  is introduced to indicate whether pair  $p$  is activated. To ensure that each IoT device is assigned to at most one pair, the following constraint is imposed

$$\sum_{p \in \mathcal{P}: k \in p} \delta_p \leq 1, \quad \forall k \in \mathcal{K}. \quad (2)$$

For an activated NOMA pair containing two devices  $p = \{k, m\}$ , the device with the stronger channel gain performs successive interference cancellation (SIC) and decodes the weaker device's signal prior to decoding its own signal. The corresponding interference structure is determined by the activated pair  $p$ .

Given the pairing configuration, the received signal-to-interference-plus-noise ratio (SINR) of IoT device  $k$  can be expressed as

$$\text{SINR}_k = \frac{p_k G_k}{\sum_{\substack{p \in \mathcal{P} \\ k, m \in p \\ m \neq k}} \delta_p p_m G_m + \sigma_{iu}^2}, \quad (3)$$

where  $p_k$  is the transmit power of device  $k$ , and  $\sigma_{iu}^2$  denotes the receiver noise power. Based on the obtained SINR, the achievable uplink data rate of device  $k$  is

$$d_k = B_{iu} \log_2(1 + \text{SINR}_k), \quad (4)$$

where  $B_{iu}$  represents the bandwidth allocated to UAV-IoT communications. Given the task data size  $D_k$ , the corresponding uplink transmission delay is

$$T_k^{tr} = \frac{D_k}{d_k}, \quad (5)$$

A predetermined visiting order is assumed for each UAV to sequentially approach all NOMA groups and independent IoT devices after its trajectory has been determined. The trajectory of UAV  $u$  is represented by the ordered set  $\{\mathbf{q}_u(0), \mathbf{q}_u(1), \dots, \mathbf{q}_u(N_u)\}$ , where  $N_u$  denotes the number of hovering waypoints assigned to UAV  $u$ . Accordingly, the total flight path length of UAV  $u$  over the entire mission is defined as

$$L_u = \sum_{n=0}^{N_u-1} \|\mathbf{q}_u(n+1) - \mathbf{q}_u(n)\|. \quad (6)$$

Given a constant flight speed  $v_f$ , the required flight time for UAV  $u$  is expressed as

$$T_u^{fly} = \frac{L_u}{v_f}. \quad (7)$$

During the data collection phase, each UAV hovers at designated waypoints to receive data from IoT devices. The hovering duration of UAV  $u$  is determined by the transmission

times of all devices within the active pairs. Based on the pair-based NOMA definition, the hovering time is expressed as

$$T_u^{hov} = \sum_{p \in P} \delta_p \max_{k \in p} \frac{D_k}{d_k}. \quad (8)$$

For NOMA pairs containing two devices, the maximum operator ensures that UAV  $u$  hovers sufficiently long to receive the data from both devices in parallel, whereas for single-device OFDMA pairs, the expression reduces to the corresponding device's transmission time.

The energy consumption of UAV  $u$  in this phase comprises both hovering energy and flight-related energy. Accordingly, the total energy consumption of the UAV-IoT subsystem is

$$E_{total} = \sum_{u=1}^U (P_h T_u^{hov} + P_f T_u^{fly}), \quad (9)$$

where  $P_h$  and  $P_f$  represent the hovering power and flight power of the UAVs, respectively, and  $T_u^{fly}$  is the flight time of UAV  $u$  as defined previously.

## 2.2. Data Offloading from UAV to LEO

After the UAV completes data collection from all IoT devices, it acquires the position information of all visible LEO satellites and selects the satellite that can satisfy the transmission demand. To represent the computation offloading decision, a binary association variable  $\beta_{s,u,t} \in \{0, 1\}$  is introduced, where  $\beta_{s,u,t} = 1$  indicates that UAV  $u$  offloads its computation to satellite  $s$  at time  $t$ . The channel gain between UAV  $u$  and LEO satellite  $s$  is modeled based on the free-space path loss as

$$G_{s,u,t} [dB] = 92.44 + 20 \log_{10}(r_{s,u,t}) + 20 \log_{10}(f_s), \quad (10)$$

where  $f_s$  denotes the operating frequency of satellite  $s$  in GHz, and  $r_{s,u,t}$  is the straight-line distance between UAV  $u$  and satellite  $s$  at  $t$ . The distance  $r_{s,u,t}$  is calculated according to the geometry of the Earth-UAV-satellite system

$$r_{s,u,t} = \sqrt{(r_e + h_s)^2 + (r_e + h_u)^2 - 2(r_e + h_s)(r_e + h_u) \cos \theta_{s,u,t}}, \quad (11)$$

where  $r_e$  is the Earth radius, and  $h_u$  and  $h_s$  are the altitudes of the UAV and LEO satellite, respectively. The satellite elevation angle  $\theta_{s,u,t}$  with respect to UAV  $u$  according to [15] is computed as

$$\theta_{s,u,t} = \arctan\left(\frac{h_s - h_u}{d_{\text{ground},s,u,t}}\right), \quad d_{\text{ground},s,u,t} = r_e \phi_{s,u,t}, \quad (12)$$

where  $d_{\text{ground},s,u,t}$  is the horizontal distance on the Earth surface between UAV  $u$  and the sub-satellite point of satellite  $s$ , and  $\phi_{s,u,t}$  denotes the corresponding central angle on the Earth's surface. The central angle  $\phi_{s,u,t}$  varies over time according to the satellite and Earth motion

$$\phi_{s,u,t} = (\omega_E \cos i - \omega_S)(t - t_0) + \phi_{s,u,t_0}, \quad (13)$$

where  $\omega_E$ ,  $\omega_S$  are the angular velocities of the Earth rotation and satellite orbit,  $i$  is the orbital inclination of the satellite,  $t_0$  is the time when the satellite becomes visible to the UAV, and  $\phi_{s,u,t_0}$  is the initial central angle at  $t_0$ .

The received power at LEO satellite  $s$  from UAV  $u$  at time  $t$  is expressed as

$$P_{s,u,t}^{re} = P_{tr} G_{tr} G_{re} G_{s,u,t}^{lin}, \quad (14)$$

where  $P_{tr}$  denotes the transmit power of the UAV,  $G_{tr}$  and  $G_{re}$  are the transmit and receive antenna gains of the UAV and LEO satellite, respectively, and  $G_{s,u,t}^{lin}$  is the linear-scale channel gain between UAV  $u$  and satellite  $s$ , derived from the distance-dependent free-space path loss as described in subsection 2.1.

Based on the received power, the achievable uplink data rate from UAV  $u$  to LEO satellite  $s$  at time  $t$  is calculated according to the Shannon capacity formula

$$d_{s,u,t} = B_{su} \log_2 \left( 1 + \frac{P_{s,u,t}^{re}}{\sigma_{su}^2} \right), \quad (15)$$

where  $B_{su}$  denotes the available bandwidth of the UAV–LEO link, and  $\sigma_{su}^2$  is the noise power at the satellite receiver. Let  $x_{s,u,t} \in \{0, 1\}$  represent the satellite association decision, where  $x_{s,u,t} = 1$  indicates that UAV  $u$  connects to satellite  $s$  at time  $t$ . To characterize unnecessary switching, the handover indicator is defined as

$$h_{u,t} = \frac{1}{2} \sum_{s=1}^S |x_{s,u,t} - x_{s,u,t-1}|. \quad (16)$$

### 2.3. Problem Formulation

To minimize the total energy consumption of the first-stage IoT–UAV data collection process, the optimization is performed over the pair-based association variables  $\delta = \{\delta_p \mid p \in P\}$ , the transmit power allocation of IoT devices  $\mathbf{p} = \{p_k \mid k \in K\}$ , and the UAV hovering positions and visiting order  $\mathbf{q} = \{\mathbf{q}_u(n) \mid u \in U\}$ . By jointly optimizing  $\delta$ ,  $\mathbf{p}$ , and  $\mathbf{q}$ , the total UAV energy consumption consisting of hovering and flight energy is minimized. The optimization problem is detailed as

$$\mathcal{P}_1 : \min_{\delta, \mathbf{p}, \mathbf{q}} \sum_{u=1}^U \left( P_h \sum_{p \in P_u} \delta_p \max_{k \in p} \frac{D_k}{d_k} + P_f \frac{L_u}{v_f} \right) \quad (17a)$$

$$\text{s.t. } p_{\min} \leq p_k \leq p_{\max}, \quad \forall k, \quad (17b)$$

$$\sum_{p: k \in p} \delta_p \leq 1, \quad \forall k, \quad (17c)$$

$$\rho p_{i(p)} G_{i(p)} \leq p_{j(p)} G_{j(p)}, \quad \forall p \in P_{\text{NOMA}}, \quad (17d)$$

$$\mathbf{q}_u(N_u) = \mathbf{q}_u(0), \quad \forall u, \quad (17e)$$

$$\delta_p \in \{0, 1\}, \quad \forall p \in P. \quad (17f)$$

Constraint 17b restricts the transmit power of each IoT device within its allowable operating range. Constraint 17c ensures that each device can participate in at most one transmission group, thereby preventing conflicting pair assignments. Constraint 17d enforces the power-domain separation required for successful NOMA decoding in every selected NOMA pair, where  $\rho \in (0, 1)$  denotes the minimum ratio ensuring the correct decoding order. Constraint 17e guarantees that each UAV completes a closed trajectory, returning to its initial hovering point after visiting all designated positions. Constraint 17f defines the binary nature of the group-selection variables.

Due to the coupling among the discrete pairing decisions  $\delta_p$ , the continuous transmit-power variables  $\{p_k\}$ , and the UAV trajectory variables  $\mathbf{q}_u(n)$ , the overall optimization problem P1 forms a mixed-integer nonlinear programming model, which is NP-hard and computationally intractable to solve optimally.

After completing the IoT data-collection stage, the UAV proceeds to offload its aggregated task data to LEO satellites. The primary objective of the second-stage satellite association problem is to minimize unnecessary handover frequency while meeting opera-

tional offloading requirements, thereby enhancing QoS assurance and indirectly reducing both transmission energy consumption and handover energy consumption for unmanned aerial vehicles.

Demand-Aware Handover does not seek to solve an optimal global objective function, but rather proposes a strategy algorithm that satisfies the following constraints

$$\mathcal{P}_2 : \min \sum_{u=1}^U \sum_{t=1}^T h_{u,t} \quad (18a)$$

$$\text{s.t.} \quad \sum_{s=1}^S \sum_{t=1}^T x_{s,u,t} d_{s,u,t} \Delta t \geq D_u, \quad (18b)$$

$$\sum_{s=1}^S x_{s,u,t} \leq 1, \quad \forall t \in \mathcal{T}, \quad (18c)$$

$$\theta_{s,u,t} \geq \theta_{\min}, \quad (18d)$$

$$\sum_{s=1}^S \sum_{t'=t}^T x_{s,u,t'} d_{s,u,t'} \Delta t \geq D_u^{\text{rem}}(t). \quad (18e)$$

Constraint 18b represents the total amount of data successfully offloaded during the mission must satisfy  $D_u$  is the task data size to be transmitted. Constraint 18c represents communication continuity requires that the UAV is associated with at most one visible satellite at any time. Constraint 18d represents only satellites satisfying the minimum elevation constraint are eligible for association. Constraint 18e ensures that switching is triggered only when the current satellite can no longer satisfy the remaining demand under predicted link evolution where  $D_u^{\text{rem}}(t)$  denotes the remaining data to be uploaded at time  $t$ .

### 3. ALGORITHM DESIGN

The model is divided into two stages to achieve an effective solution. Specifically: during the data transmission phase from IoT devices to drones, Algorithm 1 completes IoT pairing, power allocation, and drone flight path planning; during the data transmission phase from drones to low-orbit satellites, Algorithm 2 selects link-demand-aware low-orbit satellites to ensure QoS.

#### 3.1. UAV Data Collection and Energy Optimization

To effectively solve problem  $\mathcal{P}_1$ , the total energy consumption in the IoT-UAV phase is considered. Since the IoT transmission energy is negligible compared to UAV hovering and flight energy, the optimization focuses on minimizing the UAV energy consumption, which consists of hovering energy and flight energy. The problem involves coupled discrete pairing decisions  $\delta = \{\delta_p | p \in \mathcal{P}\}$ , continuous transmit power variables  $\mathbf{p} = \{p_k | k \in \mathcal{K}\}$ , and UAV hovering positions  $\mathbf{q} = \{\mathbf{q}_u(n) | u \in \mathcal{U}\}$ . This mixed-integer nonlinear programming problem is challenging to solve directly. Therefore, an algorithm framework based on alternating optimization and local search is proposed, as illustrated in Fig. 3 and detailed in Algorithm 1.

The initial NOMA pairs are constructed based on distance constraints. Specifically, the distance matrix among all IoT devices is computed, and NOMA pairs are formed by grouping the two nearest devices that satisfy the pairing constraint into the pair set  $\mathcal{P}$ . This process repeats until no new valid pairs can be formed, and the remaining devices operate as independent OFDMA nodes.

For each activated pair  $p \in \mathcal{P}$  with  $\delta_p = 1$ , an alternating optimization strategy is adopted to jointly optimize power allocation and hovering positions. Given fixed UAV



hovering position  $(x_u, y_u, h_u)$ , the transmit powers of paired devices are iteratively updated using Newton's method. For a NOMA pair  $p = \{k, m\}$ ,  $p_m$  is alternately updated while fixing  $p_k$  and vice versa until convergence. For independent OFDMA nodes,  $p_k = p_{\max}$  is directly set to maximize throughput.

Given fixed power allocation, the UAV hovering position  $(x_u, y_u)$  (with altitude  $h_u$  kept fixed) is optimized using the Adam optimizer. Unlike gradient-free methods, Adam leverages adaptive learning rates and momentum to achieve faster convergence and finer position adjustments. The position update rule is given by:

$$\mathbf{m}_t = \beta_1 \mathbf{m}_{t-1} + (1 - \beta_1) \nabla_{\mathbf{q}} E_{\text{total}}, \quad (19)$$

$$\mathbf{v}_t = \beta_2 \mathbf{v}_{t-1} + (1 - \beta_2) (\nabla_{\mathbf{q}} E_{\text{total}})^2, \quad (20)$$

$$\mathbf{q}_u^{(t+1)} = \mathbf{q}_u^{(t)} - \alpha \frac{\hat{\mathbf{m}}_t}{\sqrt{\hat{\mathbf{v}}_t + \epsilon}}, \quad (21)$$

where  $\mathbf{m}_t$  and  $\mathbf{v}_t$  are the first and second moment estimates,  $\alpha$  is the learning rate, and  $\beta_1, \beta_2$  are decay rates. The alternating optimization process iterates until reaching the maximum iteration count  $J$  or until the objective function change falls below a threshold.

After obtaining all hovering positions, the UAV visiting order must be planned to minimize flight distance. This problem is essentially a variant of the multiple traveling salesman problem (mTSP). The 2-opt local search algorithm is employed for trajectory optimization. The 2-opt algorithm iteratively removes two edges from the trajectory and reconnects them to eliminate path crossings, thereby progressively improving solution quality until reaching a local optimum. Compared to metaheuristic methods, 2-opt provides faster convergence and more stable performance for trajectory optimization.

To further reduce energy consumption, a pair recombination mechanism is introduced. All unpaired nodes are traversed to check if there exist opportunities to recombine with existing paired nodes. If an unpaired node is closer to one member of an existing pair, the original pair is broken and a new NOMA pair is formed. For two spatially close NOMA pairs, their members are exchanged to reduce UAV flight distance or hovering time.

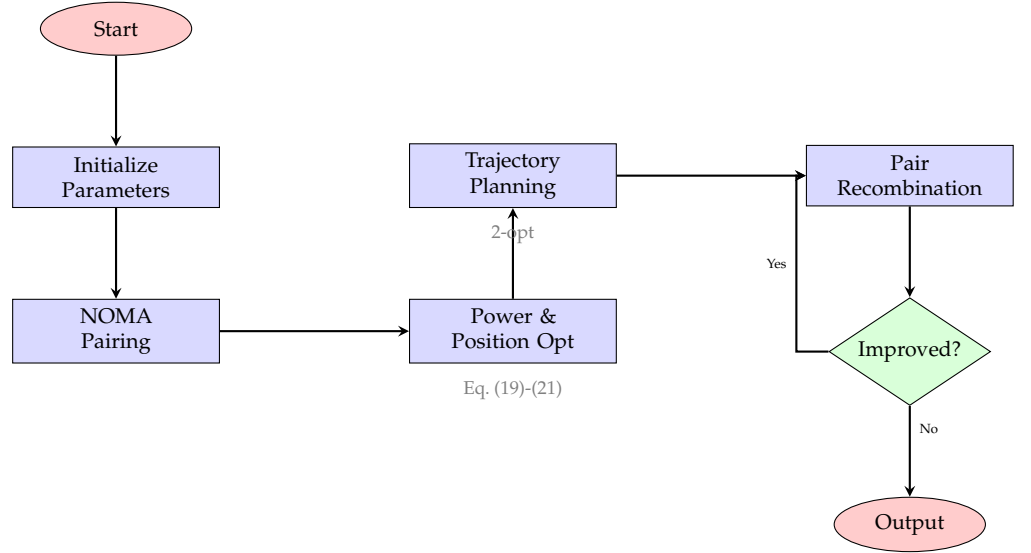
After each recombination or exchange operation, the UAV trajectory and total energy consumption are recalculated. The new configuration is accepted if it yields lower energy consumption; otherwise, the original configuration is retained. Through multiple rounds of exchange operations, the algorithm can escape local optima and obtain better NOMA pairing schemes. The complete algorithm flow is presented in Algorithm 1 with the corresponding flowchart shown in Fig. 3.

### 3.2. LEO Satellite Selection Optimization

After the UAV completes data collection from all IoT devices, it must offload the aggregated data to LEO satellites for further processing. Unlike static ground stations, LEO satellites exhibit high mobility with time-varying channel conditions and limited visibility windows. Consequently, the satellite association decision directly impacts both transmission efficiency and handover frequency. Frequent satellite handovers introduce signaling overhead, packet loss, and interruption delays, degrading quality of service. Conversely, maintaining connection with a suboptimal satellite prolongs transmission time and increases energy consumption. Therefore, the selection strategy must balance handover frequency against link quality to ensure timely and reliable data offloading.

To address problem P2, a demand-aware satellite selection mechanism is proposed. Rather than myopically selecting the satellite with maximum instantaneous throughput at each time slot, the algorithm evaluates whether the current satellite can satisfy the remaining data transmission demand under predicted link evolution. Satellite handover is triggered only when the current satellite is predicted to be insufficient for completing the





**Figure 3.** Flowchart of the proposed algorithm for UAV data collection and energy optimization.

**Table 1.** SIMULATION PARAMETERS

Parameter	Value	Parameter	Value
$K$ (IoT devices)	20	$N$ (hover points)	10
$D_k$ (data size)	10 MB	$B_{iu}$	1 MHz
$P_{\max}$	5 W	$P_{\min}$	0.1 W
$P_h$	100 W	$P_f$	150 W
$v_f$	10 m/s	$h_u$	100 m
$\beta_0$	$10^{-5}$	$\sigma_{iu}^2$	$10^{-9}$ W
$\rho$	0.8	$d_{\max}$	100 m
Area size	$500 \times 500 \text{ m}^2$	$\alpha$ (Adam)	0.01
$\beta_1$ (Adam)	0.9	$\beta_2$ (Adam)	0.999
$J$ (max iterations)	100	$\epsilon$	$10^{-6}$

**Table 2.** LEO SATELLITE PARAMETERS

Parameter	Value	Parameter	Value
$\theta_{\min}$	$15^\circ$	$B_{su}$	10 MHz
$P_{tr}$	10 W	$G_{tr}$	10 dBi
$G_{re}$	30 dBi	$\sigma_{su}^2$	$4 \times 10^{-14}$ W
$f_s$	20 GHz	$r_e$	6378 km
$h_s$ (LEO altitude)	550 km	$\omega_E$	7.29 rad/s
Constellation	Starlink	Num. satellites	200

remaining task, thereby minimizing unnecessary switching while ensuring transmission deadlines. 254  
255

The algorithm operates as follows. At each decision epoch, all LEO satellites satisfying 256  
the elevation angle constraint  $\theta_{s,u,t} \geq \theta_{\min}$  are identified as candidate satellites. For each 257  
candidate  $s$ , the instantaneous uplink data rate  $d_{s,u,t}$  is calculated according to Eq. (15) 258  
based on the distance-dependent channel gain and transmit power. The algorithm then 259  
predicts the cumulative throughput achievable by the current satellite over the remaining 260  
visibility window. If the predicted throughput exceeds the remaining data size  $D_u^{\text{rem}}(t)$ , the 261  
current satellite association is maintained. Otherwise, the algorithm selects the satellite 262  
with maximum instantaneous throughput from the candidate set and performs handover. 263

**Algorithm 1** UAV Data Collection and Energy Optimization

---

```

1: Input: All IoT devices' positions  $\mathbf{q}_k$  and corresponding packets  $D_k$ , UAV initial position  $\mathbf{q}_u(0)$ 
2: Output: Energy consumption  $E_{\text{total}}$ 
3: Initialize distance matrix and maximum iterations  $J$ 
4: Form NOMA pairs from two nearest devices under distance constraints
5: Obtain lists of paired and unpaired nodes
6: for each paired nodes  $(k, m)$  do
7:   Initialize transmission powers  $p_k, p_m$ 
8:   Initialize hover position  $(x_u, y_u, h_u)$ 
9:   Set iteration counter  $j \leftarrow 0$ 
10:  while  $j \leq J$  do
11:    Update  $p_k^{(j)}, p_m^{(j)}$  using Newton's method
12:    Update  $(x_u^{(j)}, y_u^{(j)}, h_u^{(j)})$  using Adam optimizer
13:     $j \leftarrow j + 1$ 
14:  end while
15: end for
16: Optimize UAV trajectories using 2-opt local search
17: Compute energy consumption  $E_{\text{total}}$  using Eq. (9)
18: for each node combination (unpaired and paired) do
19:   Exchange nodes between paired and unpaired groups
20:   Update UAV trajectories and device transmission powers
21:   Calculate new energy consumption  $E_{\text{new}}$  using Eq. (9)
22:   if  $E_{\text{new}} < E_{\text{total}}$  then
23:      $E_{\text{total}} \leftarrow E_{\text{new}}$ 
24:   end if
25: end for
26: return  $E_{\text{total}}$ 

```

---

**4. SIMULATION RESULTS**

This section evaluates the performance of our proposed two-stage optimization framework for UAV-assisted IoT data collection and LEO satellite offloading in SAGIN. We begin by presenting the simulation setup for both algorithms, followed by comprehensive performance analyses that demonstrate the effectiveness of each stage. The evaluation encompasses not only individual algorithm performance but also the synergistic effects of the end-to-end system.

**4.1. Simulation Setup**

Our evaluation framework is designed to assess a complete data collection and offloading mission in SAGIN, where ground IoT devices transmit data to UAVs (Stage 1), and UAVs subsequently offload the aggregated data to LEO satellites for cloud processing (Stage 2). All simulations are implemented in Python 3.10 and executed on a workstation equipped with an Intel Core i7-12700K processor and 32GB RAM.

The first stage focuses on the ground-to-air segment, where UAVs must efficiently collect data from spatially distributed IoT devices while minimizing energy consumption. We consider a square coverage area of  $500 \text{ m} \times 500 \text{ m}$  containing  $K$  IoT devices, where  $K$  varies from 20 to 50 to evaluate scalability. Three UAVs operate at a fixed altitude of 100 m with a maximum transmit power of 0.1 W per device. The communication channel is modeled with a bandwidth of 1 MHz and noise power spectral density of  $-174 \text{ dBm/Hz}$ . The path loss exponent is set to 2.5, representing typical urban propagation environments. To ensure statistical reliability, each configuration is evaluated across three different random seeds (27, 37, 41), and results are averaged. The UAV hovering power and flight power are set to  $P_h = 80 \text{ W}$  and  $P_f = 120 \text{ W}$ , respectively, based on typical small UAV specifications.

**Algorithm 2** Demand-Aware LEO Satellite Selection

---

```

1: Input: UAV position  $\mathbf{q}_u(t)$ , total data size  $D_u$ , transmitted data  $D_u^{\text{tx}}$ , current satellite
    $s_{\text{curr}}$ , time  $t$ , prediction horizon  $\Delta t$ 
2: Output: Selected satellite  $s^*$ 
3: Compute remaining data:  $D_u^{\text{rem}}(t) \leftarrow D_u - D_u^{\text{tx}}$ 
4: Initialize candidate set:  $\mathcal{S}_{\text{candidate}} \leftarrow \emptyset$ 
5: for each satellite  $s \in \mathcal{S}$  do
6:   Compute elevation angle  $\theta_{s,u,t}$  using Eq. (12)
7:   if  $\theta_{s,u,t} \geq \theta_{\min}$  then
8:     Add  $s$  to  $\mathcal{S}_{\text{candidate}}$ 
9:   end if
10: end for
11: if  $s_{\text{curr}} \in \mathcal{S}_{\text{candidate}}$  then
12:   Compute current throughput:  $d_{s_{\text{curr}},u,t}$  using Eq. (15)
13:   Predict remaining visibility time:  $T_{\text{vis}} \leftarrow T_{\text{access}}(s_{\text{curr}}) - t$ 
14:   Estimate achievable data:  $\hat{D}_{\text{curr}} \leftarrow d_{s_{\text{curr}},u,t} \cdot \min(T_{\text{vis}}, \Delta t)$ 
15:   if  $\hat{D}_{\text{curr}} \geq D_u^{\text{rem}}(t)$  then
16:     return  $s_{\text{curr}}$  ▷ Current satellite sufficient
17:   end if
18: end if
19:  $s^* \leftarrow \arg \max_{s \in \mathcal{S}_{\text{candidate}}} d_{s,u,t}$  ▷ Select maximum throughput satellite
20: return  $s^*$ 

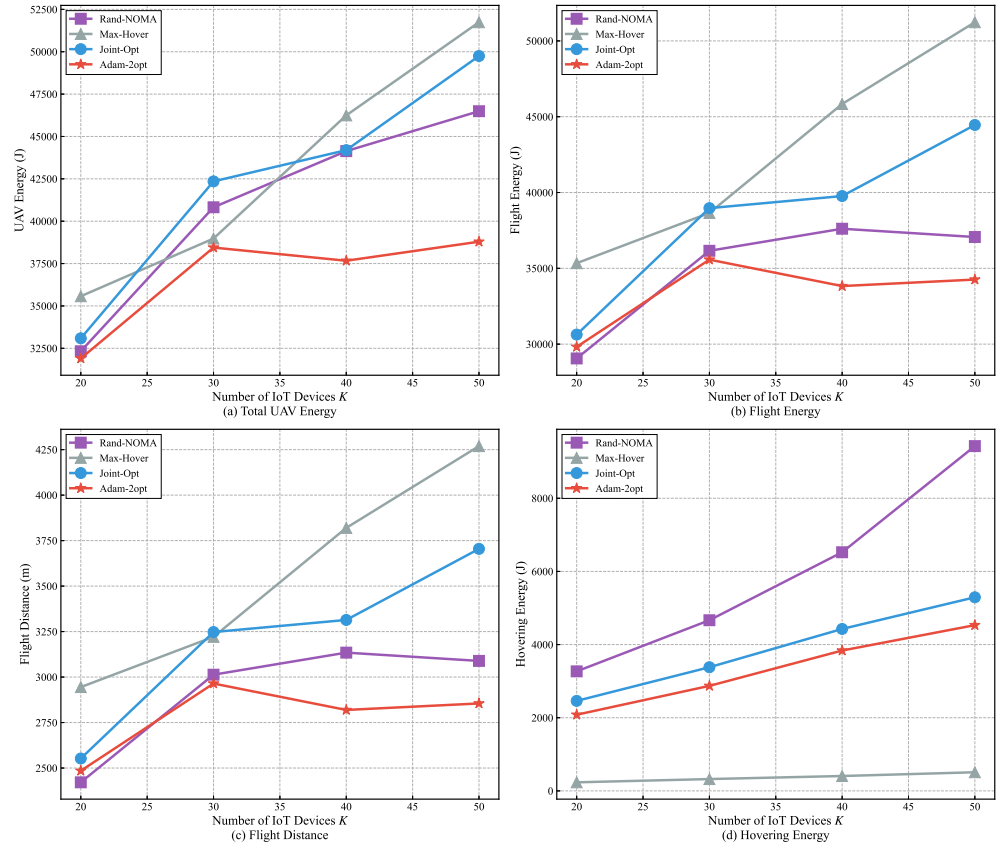
```

---

To benchmark the proposed Algorithm 1, we compare against three baseline methods representing different optimization strategies. The *Random Pairing* baseline randomly groups IoT devices into NOMA pairs without considering spatial proximity or channel conditions, establishing a lower bound on achievable performance. The *Fixed Hovering* method employs a greedy strategy that maximizes hovering time at each service point but neglects trajectory optimization, resulting in potentially long flight distances. The *Basic Optimization* approach incorporates NOMA pairing, power allocation, and position optimization but lacks the advanced trajectory refinement of our proposed method. Our *Proposed Method* (denoted as Adam-2opt) builds upon Basic Optimization by integrating the Adam optimizer for hovering position refinement and applying 2-opt local search for trajectory optimization.

The second stage addresses the air-to-space segment, where UAVs must maintain stable connectivity with rapidly moving LEO satellites. We simulate the Starlink constellation at an altitude of 550 km, with the number of visible satellites varying from 500 to over 1500 depending on the time window. The UAV is positioned at (15°N, 118°E) at an altitude of 200 m, using a carrier frequency of 20 GHz and channel bandwidth of 10 MHz. The minimum elevation angle is constrained to  $\theta_{\min} = 10^\circ$  to ensure acceptable link quality. Each handover incurs a 400 ms delay during which no data transmission occurs, representing realistic RF reconfiguration and protocol negotiation overhead. The simulation duration spans 10 minutes with a time step of 20 seconds, allowing us to observe multiple satellite passes and handover events. The demand rate is configured to 8 Mbps, corresponding to a typical 1080p video streaming application that represents moderate data offloading requirements.

We compare our proposed *Demand-Aware* handover strategy against a conventional *Greedy* approach. The Greedy method continuously seeks the satellite offering maximum instantaneous data rate, switching whenever a better candidate becomes visible. This strategy maximizes throughput but potentially causes frequent handovers and service interruptions. In contrast, our Demand-Aware method maintains the current satellite connection as long as it can satisfy the remaining data transmission requirements, switching only when



**Figure 4.** Performance comparison of four algorithms across varying numbers of IoT devices: (a) Total UAV energy consumption, (b) Flight energy, (c) Flight distance, and (d) Hovering energy. The proposed Adam-2opt method consistently achieves the lowest total energy consumption across all scenarios.

necessary to meet the offloading deadline. This philosophy prioritizes connection stability and quality of service over peak throughput maximization.

#### 4.2. Performance of IoT-UAV Data Collection Stage

The IoT-UAV data collection stage presents a challenging mixed-integer nonlinear programming problem involving coupled decisions on device pairing, power allocation, and UAV trajectory planning. Fig. 4 presents a comprehensive performance comparison across four key metrics: total UAV energy consumption, flight energy, flight distance, and hovering energy. These results reveal several critical insights into the effectiveness of our proposed optimization approach.

Examining the total UAV energy consumption in Fig. 4(a), we observe that the proposed Adam-2opt algorithm consistently outperforms all baseline methods across the entire range of IoT device densities. For the most demanding scenario with  $K = 50$  devices, Adam-2opt achieves a total energy consumption of 38,789.84 J, representing a 16.6% reduction compared to Random Pairing (46,490.88 J), a 25.0% reduction compared to Fixed Hovering (51,741.77 J), and a 22.0% reduction compared to Basic Optimization (49,747.93 J). The improvement becomes more pronounced as the network scales, with only 1.3% energy savings at  $K = 20$  but expanding to 16.6% at  $K = 50$ . This trend indicates that the benefits of sophisticated trajectory optimization become increasingly significant in dense IoT deployments, where the complexity of visit sequencing and path planning grows substantially.

The superior energy performance of Adam-2opt stems from its effectiveness in both spatial dimensions of the optimization problem. Fig. 4(c) demonstrates that our method achieves significantly shorter flight distances compared to all baselines. At  $K = 50$ , Adam-2opt requires only 2,854.92 m of flight distance, which is 7.6% shorter than Random Pairing (3,088.77 m), 21.9% shorter than Basic Optimization (3,704.63 m), and 33.1% shorter than Fixed Hovering (4,269.38 m). The 2-opt local search algorithm plays a crucial role here by systematically eliminating trajectory crossings and unnecessary detours. This trajectory refinement not only reduces flight distance but also decreases flight energy proportionally, as evidenced in Fig. 4(b), where Adam-2opt consumes 34,259.07 J of flight energy compared to 37,065.30 J for Random Pairing and 51,232.53 J for Fixed Hovering.

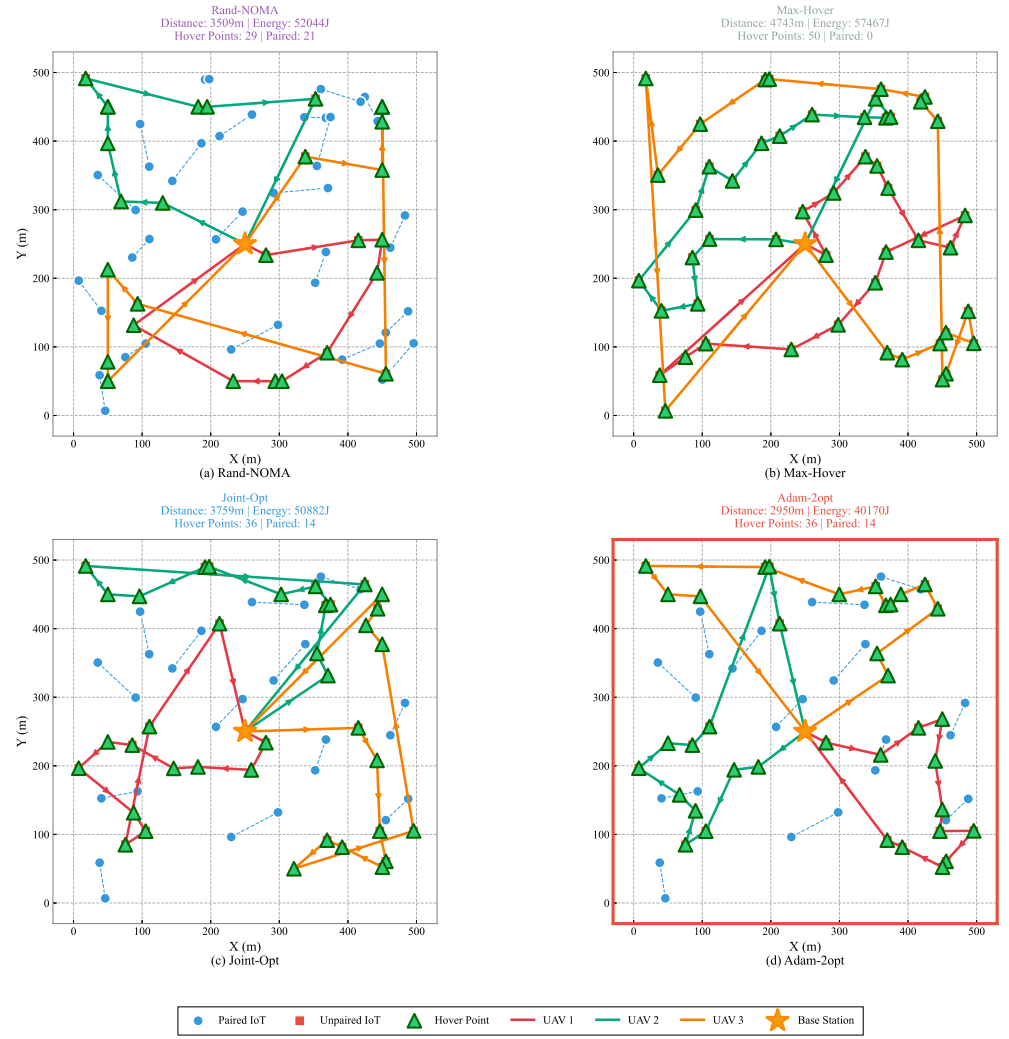
An interesting observation emerges when examining the hovering energy patterns in Fig. 4(d). Fixed Hovering exhibits the lowest hovering energy (509.25 J at  $K = 50$ ) by design, as it minimizes the time spent at each service location. However, this strategy forces UAVs to traverse longer distances between widely separated service points, resulting in the highest total energy consumption. This demonstrates a fundamental trade-off in UAV mission planning: minimizing hovering time alone is insufficient if it leads to excessive flight distance. The Random Pairing baseline suffers from the opposite problem, with hovering energy of 9,425.58 J at  $K = 50$ —more than double that of Adam-2opt (4,531.19 J). This inefficiency arises from suboptimal device pairing that creates geographically dispersed clusters, forcing UAVs to hover longer to serve poorly matched NOMA pairs. Our proposed method strikes an optimal balance, achieving 51.9% hovering energy reduction compared to Random Pairing while simultaneously maintaining minimal flight distance.

To better understand the spatial efficiency of each algorithm, Fig. 5 visualizes the actual UAV trajectories for a representative scenario with  $K = 50$  devices. The visual comparison starkly illustrates the limitations of baseline approaches and the advantages of our optimization framework. The Random Pairing trajectory in Fig. 5(a) exhibits numerous inefficiencies: UAV paths cross multiple times, backtracking occurs frequently, and the hovering points are distributed without clear spatial logic. This chaotic pattern reflects the arbitrary pairing decisions that ignore geographical proximity. The trajectory spans 3,509 m with 29 hovering points serving 21 paired devices, indicating substantial wasted motion.

The Fixed Hovering approach in Fig. 5(b) presents a different pathology. While the hovering points are well-distributed across the coverage area (50 points to ensure comprehensive coverage), the predetermined nature of these locations forces UAVs to visit many unnecessary waypoints. The resulting trajectory is remarkably long (4,743 m), with extensive empty flight segments between service clusters. This inflexibility in hovering point placement represents a fundamental flaw: maximizing hover point density does not translate to mission efficiency when device locations are non-uniform.

Basic Optimization in Fig. 5(c) shows noticeable improvement over the two baselines. The algorithm successfully optimizes hovering point positions to better match device clusters, reducing the trajectory length to 3,759 m. However, the trajectory still contains visible crossings and suboptimal turn sequences, indicating that position optimization alone is insufficient. The lack of trajectory-level refinement leaves opportunities for further improvement.

The proposed Adam-2opt method in Fig. 5(d) exhibits the most efficient trajectory structure. The path is notably smoother, with minimal backtracking and no trajectory crossings. The 2-opt refinement has successfully reordered the visit sequence to create a more logical flow pattern, reducing the total trajectory length to 2,950 m—the shortest among all methods. Importantly, this is achieved with 36 hovering points serving 14 paired devices, demonstrating that our algorithm balances multiple objectives simultaneously. The trajectory forms a more circular pattern that efficiently covers the device distribution while



**Figure 5.** Trajectory visualization comparison for four algorithms ( $K = 50$  devices, seed 27): (a) Random Pairing exhibits random hovering point placement with multiple trajectory crossings, (b) Fixed Hovering creates extended flight paths due to predetermined service locations, (c) Basic Optimization shows improved hovering point placement through position optimization, and (d) Adam-2opt (Proposed) demonstrates the shortest and most efficient trajectories with eliminated crossings and optimal hovering point distribution.

minimizing redundant motion. Each segment serves a clear purpose, and the directional flow minimizes the angular changes that would otherwise consume additional flight time and energy.

Beyond the primary energy and distance metrics, the computational complexity of each algorithm warrants consideration for practical deployment scenarios. The Random Pairing method executes in merely 0.1064 seconds for  $K = 50$  devices, as it performs only simple pairing operations without optimization. Fixed Hovering completes even faster at 0.0045 seconds due to its predetermined hovering grid and greedy power allocation. Basic Optimization requires 0.0613 seconds, balancing computational overhead with optimization quality. The proposed Adam-2opt method incurs the highest computational cost at 1.2024 seconds, attributable to the iterative Adam optimization and 2-opt trajectory refinement. However, this overhead remains entirely acceptable for offline mission planning scenarios, where trajectories can be pre-computed before deployment. The 16.6% energy savings achieved by Adam-2opt far outweigh the modest increase in planning time, especially

considering that UAV battery capacity is the primary limiting factor in long-duration missions.

The scalability characteristics revealed in Fig. 4 provide important insights for system design. At low device densities ( $K = 20$ ), all methods achieve relatively similar performance (within 11% of each other), suggesting that sophisticated optimization is less critical when the problem complexity is limited. However, as the network scales to  $K = 50$ , the performance gap widens dramatically. Fixed Hovering's energy consumption increases by 45% from  $K = 20$  to  $K = 50$ , while Adam-2opt increases by only 21.6%. This superior scaling behavior indicates that our algorithm's benefits become more pronounced precisely when they are most needed—in large-scale IoT deployments. The sublinear growth in Adam-2opt's energy consumption suggests that the algorithm effectively exploits the spatial clustering opportunities that emerge in denser networks, grouping nearby devices into efficient service routes.

#### 4.3. Performance of UAV-LEO Offloading Stage

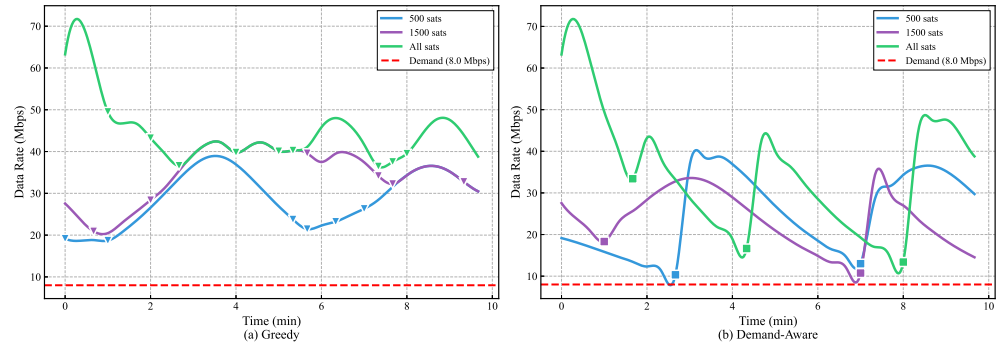
Having established that Algorithm 1 effectively minimizes UAV energy consumption during ground data collection, we now examine the second stage of the system: offloading the aggregated data from UAVs to LEO satellites. This stage introduces distinct challenges fundamentally different from those in the IoT-UAV segment. LEO satellites orbit at approximately 7 km/s, resulting in rapidly time-varying channel conditions and limited visibility windows typically lasting only a few minutes. While Algorithm 1 focuses on spatial optimization in a relatively static environment, Algorithm 2 must address temporal dynamics and maintain service continuity despite continuous satellite motion.

The primary challenge in this stage is not maximizing instantaneous throughput, but rather balancing throughput requirements with connection stability. Frequent satellite handovers degrade quality of service through packet loss during switching intervals, increased signaling overhead, and potential service interruptions. Our Demand-Aware strategy addresses this challenge by maintaining the current satellite connection as long as it satisfies the data offloading requirements, switching only when the predicted link quality becomes insufficient to complete the remaining transmission within the mission deadline.

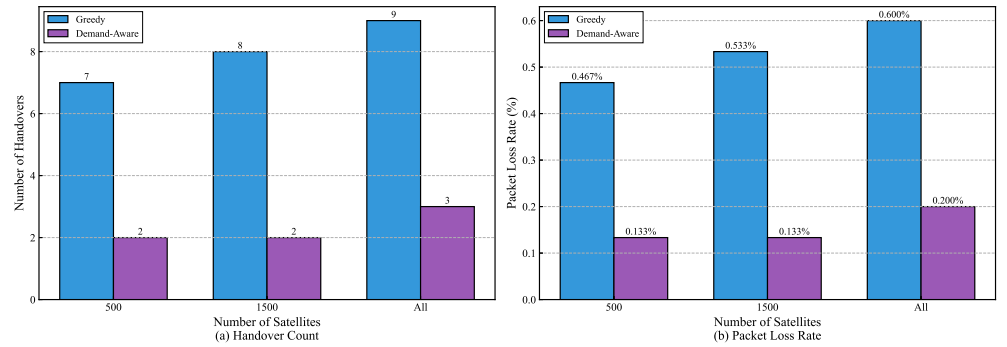
Fig. 6 illustrates the data rate evolution over a 10-minute observation window for both the Greedy and Demand-Aware strategies across three satellite density scenarios. The contrast in behavior is immediately apparent. The Greedy strategy exhibits highly volatile data rates with frequent fluctuations between peaks above 40 Mbps and valleys below 20 Mbps. The numerous handover markers (triangles for Greedy, squares for Demand-Aware) clustered throughout the timeline reveal aggressive switching behavior. In the 500-satellite scenario, Greedy performs 7 handovers, while in the full constellation scenario, this increases to 9 handovers. Each handover represents a potential service disruption and contributes to packet loss during the 400 ms switching interval.

In contrast, the Demand-Aware strategy in Fig. 6(b) demonstrates remarkably stable behavior. The data rates remain consistently above the 8 Mbps demand threshold (indicated by the red dashed line) throughout the entire mission, with only 2-3 handovers across all scenarios. The rate stability is particularly evident in the 500-satellite case, where Demand-Aware maintains smooth rate curves with minimal fluctuations despite having fewer satellites available. This stability arises from the algorithm's conservative switching policy: as long as the current satellite can deliver sufficient throughput to complete the remaining data transmission within the available time window, no handover is triggered. Only when the predicted future capacity becomes inadequate does the algorithm proactively switch to a better candidate.





**Figure 6.** Data rate comparison between Greedy and Demand-Aware strategies across different satellite densities: (a) Greedy strategy exhibits frequent rate fluctuations and numerous handovers (marked with triangles), pursuing maximum instantaneous throughput, while (b) Demand-Aware strategy maintains stable rates above the 8 Mbps demand threshold (red dashed line) with significantly fewer handovers (marked with squares), switching only when necessary to meet transmission requirements.



**Figure 7.** Handover frequency and packet loss comparison: (a) Demand-Aware achieves 66.7–75.0% handover reduction across all satellite densities, with the most significant improvement (75.0%) in the 1500-satellite scenario, (b) Corresponding packet loss rates show proportional reduction, with Demand-Aware maintaining remarkably low loss rates of 0.133–0.200% compared to 0.467–0.600% for Greedy.

The quantitative handover analysis in Fig. 7(a) reveals the dramatic difference in switching behavior. For the 500-satellite scenario, Demand-Aware reduces handovers from 7 (Greedy) to just 2, representing a 71.4% reduction. This improvement persists across all constellation densities: 75.0% reduction for 1500 satellites (from 8 to 2) and 66.7% reduction for the full constellation (from 9 to 3). Notably, the Demand-Aware handover count remains remarkably stable regardless of satellite density—consistently between 2 and 3—while the Greedy count increases with constellation size as more switching opportunities become available. This stability demonstrates that Demand-Aware bases decisions on transmission requirements rather than merely responding to instantaneous channel conditions.

The reduction in handover frequency directly translates to improved packet delivery performance, as shown in Fig. 7(b). The packet loss rate is calculated based on the 400 ms handover delay: each handover causes a 400 ms service interruption during which transmitted packets are lost. For a 10-minute mission, the packet loss rate is computed as  $(\text{Number of Handovers} \times 400 \text{ ms}) / (600 \text{ s} \times 1000 \text{ ms}) \times 100\%$ . In the 500-satellite scenario, Demand-Aware achieves a packet loss rate of only 0.133%, compared to 0.467% for Greedy—a 71.4% reduction. The most dramatic improvement occurs in the 1500-satellite case, where packet loss drops from 0.533% to 0.133%, representing a 75.0% reduction. These seemingly small percentages translate to significant practical impact: at an 8 Mbps data rate, a 0.533% loss rate means approximately 42.6 kbit of lost data over 10 minutes, compared to

just 10.6 kbit for Demand-Aware. For latency-sensitive applications such as real-time video streaming or industrial control, this reliability improvement is crucial.

An important trade-off emerges when comparing average data rates. The Greedy strategy achieves higher average throughput: 28.62 Mbps for 500 satellites, 34.29 Mbps for 1500 satellites, and 44.78 Mbps for the full constellation. In comparison, Demand-Aware obtains 24.87 Mbps, 23.37 Mbps, and 35.27 Mbps, respectively—representing 13.1% to 31.8% lower average rates. However, this throughput sacrifice must be evaluated in context. The demand rate is set to 8 Mbps, and Demand-Aware consistently delivers  $3\times$  to  $4\times$  this requirement. The excess capacity provides a comfortable margin for handling temporary channel degradation without triggering handovers. Meanwhile, Greedy's pursuit of maximum throughput comes at the cost of 200–300% higher packet loss and more frequent service interruptions. For most applications, meeting the demand threshold reliably is more valuable than maximizing peak throughput at the expense of stability.

The scalability characteristics across constellation densities reveal contrasting behaviors. Greedy's handover count increases moderately with satellite density (+28.6% from 500 to full constellation), as more satellites create more switching opportunities. This trend is problematic because future LEO deployments will feature increasingly dense constellations, potentially exacerbating the handover problem. Conversely, Demand-Aware maintains consistently low handover counts (2–3) regardless of density, demonstrating excellent scalability. The algorithm's switching decisions depend on whether current capacity can meet remaining demand, not on how many alternative satellites are visible. This fundamental design difference makes Demand-Aware naturally robust to constellation density variations.

The performance in the 500-satellite scenario deserves particular attention as it represents the optimal operating point for Demand-Aware. With only 2 handovers and a packet loss rate of just 0.133%, the system achieves exceptional stability while maintaining an average rate of 24.87 Mbps—more than sufficient for the 8 Mbps demand. This suggests that moderate satellite density may be preferable to maximum density for stability-critical applications. The 1500-satellite scenario also performs excellently with identical handover count (2) and packet loss (0.133%), indicating that Demand-Aware can leverage increased satellite availability to maintain quality without triggering excessive switching.

#### 4.4. End-to-End System Analysis

The preceding subsections have demonstrated the effectiveness of Algorithm 1 in minimizing UAV energy consumption during IoT data collection and Algorithm 2 in reducing handover frequency during satellite offloading. However, the true value of our framework lies in the synergistic integration of these two stages into a cohesive end-to-end system. To fully appreciate this synergy, we now examine how the algorithms work together to optimize overall mission performance from ground-based IoT devices to cloud-based processing infrastructure.

Consider a representative mission scenario with  $K = 50$  IoT devices and 500 visible LEO satellites. In the first stage, the proposed Adam-2opt algorithm collects data from all IoT devices using optimized trajectories, consuming 38,789.84 J of UAV energy over a flight distance of 2,854.92 m. The data collection completes with 25 hovering points serving 22 paired devices, taking approximately  $T_{\text{collect}} = T_{\text{hover}} + T_{\text{flight}} = 56.64 + (2854.92/10) = 342.13$  s, assuming a typical UAV flight speed of 10 m/s. At this point, the UAVs have aggregated the IoT data and are ready to begin the second stage.

In the second stage, the UAVs offload their aggregated data to LEO satellites using the Demand-Aware handover strategy. With an average data rate of 24.87 Mbps and only 2 handovers occurring over the 10-minute observation window, the system achieves stable

**Table 3.** End-to-End System Performance Comparison ( $K = 50$ , 500 Satellites)

Metric	Baseline	Proposed	Improvement
UAV Energy (J)	46,490.88	38,789.84	−16.6%
Flight Distance (m)	3,088.77	2,854.92	−7.6%
Collection Time (s)	426.70	342.13	−19.8%
Handover Count	7	2	−71.4%
Packet Loss (%)	0.467	0.133	−71.4%
Avg. Data Rate (Mbps)	28.62	24.87	−13.1%

Baseline: Random Pairing + Greedy; Proposed: Adam-2opt + Demand-Aware

connectivity with minimal service interruptions. The packet loss rate of 0.133% ensures that virtually all collected data successfully reaches the cloud. Comparing this integrated performance against a baseline system using Random Pairing for Stage 1 and Greedy for Stage 2 reveals the comprehensive advantages of our approach: 16.6% reduction in UAV energy consumption, 71.4% reduction in satellite handovers, and 71.4% reduction in packet loss—all achieved simultaneously.

The key insight is that optimizing each stage independently is insufficient; the stages must be designed with complementary objectives. Algorithm 1 focuses on spatial efficiency (minimizing flight distance and hovering time) while Algorithm 2 emphasizes temporal stability (minimizing switching frequency). These objectives are naturally compatible: efficient data collection reduces the total data volume that must be offloaded, which in turn reduces the transmission time required and allows the Demand-Aware strategy to maintain stable connections with fewer handovers. Conversely, stable satellite connectivity reduces the risk of transmission failures that would otherwise require retransmission and additional UAV energy expenditure.

Table 3 summarizes the end-to-end performance metrics for the complete system. The proposed two-stage framework achieves substantial improvements across all dimensions compared to baseline combinations. The 16.6% UAV energy reduction directly extends mission duration and operational range, critical factors for battery-constrained UAVs. The 71.4% handover reduction minimizes service interruptions and signaling overhead, which is particularly valuable for latency-sensitive applications. The 71.4% packet loss reduction ensures reliable data delivery even under dynamic channel conditions. While the average data rate is 13.1% lower than the Greedy approach, this trade-off is well-justified given that the achieved rate (24.87 Mbps) exceeds the demand requirement (8 Mbps) by a factor of  $3\times$ , providing ample margin for handling temporary channel degradation.

The scalability analysis across varying IoT device densities reveals interesting trends in end-to-end performance. At small network sizes ( $K = 20$ ), the energy advantage of Algorithm 1 is modest (1.3% improvement), as baseline methods already perform reasonably well when the problem complexity is limited. However, the handover reduction from Algorithm 2 remains substantial (71.4%) even at this scale, indicating that the satellite selection problem is inherently challenging regardless of ground network size. As the network scales to  $K = 50$ , the energy savings increase dramatically to 16.6%, demonstrating that sophisticated trajectory optimization becomes increasingly valuable in dense deployments. Importantly, the handover reduction remains stable across all device densities, confirming that Algorithm 2's performance is robust to variations in the ground segment workload. This independent scalability of the two stages is a desirable property: Algorithm 1 adapts to ground network complexity while Algorithm 2 maintains consistent performance in the space segment.

The robustness of the system to satellite density variations further validates our design approach. While Greedy’s handover count increases with constellation size (from 7 to 9 as satellites increase), Demand-Aware maintains consistently low handover counts (2–3) across all densities. This behavior ensures that the system will perform well even as LEO constellations continue to expand in future deployments. The proposed framework demonstrates a fundamental principle: satisfying operational requirements with minimal resource consumption is often superior to maximizing instantaneous performance metrics at the cost of stability and efficiency.

## 5. Discussion

The simulation results presented in this section reveal several important insights into the design of efficient UAV-assisted IoT data collection and LEO satellite offloading systems. Beyond the quantitative performance improvements, the experiments illuminate fundamental principles that should guide future system designs in SAGIN environments.

The success of Algorithm 1 demonstrates that effective UAV trajectory optimization requires simultaneous consideration of multiple coupled objectives. Optimizing only hovering positions, as in Basic Optimization, yields suboptimal results because it ignores the flight path structure. Similarly, minimizing hovering time alone, as in Fixed Hovering, leads to excessive flight distance. Our Adam-2opt approach achieves superior performance by recognizing that these objectives are fundamentally interrelated: the optimal hovering positions depend on the trajectory that connects them, and the optimal trajectory depends on where the hovering points are located. The Adam optimizer provides fine-grained position adjustments through gradient-based updates, while 2-opt refines the trajectory structure through combinatorial search. This hybrid approach combines the strengths of continuous optimization and discrete search, yielding better solutions than either technique alone.

The Demand-Aware handover strategy embodies a paradigm shift in satellite network optimization. Traditional approaches, exemplified by the Greedy baseline, focus on maximizing instantaneous performance metrics such as data rate or signal strength. This myopic perspective fails to account for the long-term consequences of frequent switching: handover delays, packet loss, signaling overhead, and degraded user experience. Our Demand-Aware strategy adopts a fundamentally different philosophy: determine the minimum requirements to complete the mission successfully, and avoid unnecessary actions that would jeopardize stability. This principle of *sufficiency-based optimization*—doing just enough to meet the objective without over-optimizing secondary metrics—proves remarkably effective. By maintaining connections that satisfy demand requirements rather than chasing peak throughput, the algorithm achieves 66.7–75.0% handover reduction while still delivering  $3\times$  the required data rate.

The synergy between the two algorithms highlights a critical insight for multi-stage system design: the objectives optimized in different stages should be complementary rather than competing. Algorithm 1 minimizes UAV energy by reducing spatial inefficiencies (flight distance, hovering time), while Algorithm 2 minimizes handover overhead by reducing temporal instability (switching frequency, service interruptions). These objectives align naturally: energy-efficient data collection reduces the transmission burden in the offloading stage, and stable satellite connectivity reduces the risk of retransmission that would otherwise waste UAV energy. This complementary design ensures that optimizing one stage does not inadvertently degrade the other.

Comparing our approach to potential alternatives reveals important trade-offs. One might ask: why not use a single integrated optimization that jointly optimizes UAV trajectories and satellite handovers? While theoretically appealing, such an approach would

face severe computational challenges due to the vastly different timescales and dynamics of the two stages. UAV trajectory planning operates on minutes-to-hours timescales with relatively static device positions, allowing for offline pre-computation. Satellite handover decisions must be made in real-time (seconds) based on rapidly changing orbital dynamics. Combining these problems would either force the trajectory optimizer to account for unpredictable future satellite positions (intractable) or require the handover algorithm to wait for trajectory recomputation (impractical). Our two-stage approach respects these temporal separations while still achieving strong overall performance.

The trade-off between energy efficiency and computational complexity also merits discussion. The proposed Adam-2opt algorithm requires 1.2024 seconds of computation time for  $K = 50$  devices, significantly longer than the 0.0613 seconds needed by Basic Optimization. However, this overhead is well-justified in the context of offline mission planning. UAV missions are typically planned hours or days in advance, and a few additional seconds of planning time is negligible compared to the mission duration (typically 30–60 minutes). The 16.6% energy savings translate directly to extended mission range or longer operational time, both of which are far more valuable than saving one second of pre-flight computation. For applications requiring real-time replanning due to dynamic events (sudden device failures, obstacle avoidance), faster approximation algorithms could be substituted, though at some cost to solution quality.

The observed scalability trends suggest that our approach becomes increasingly valuable as SAGIN systems expand. The 16.6% energy improvement at  $K = 50$  devices, compared to just 1.3% at  $K = 20$ , indicates that sophisticated optimization techniques are most beneficial precisely when they are most needed—in large-scale deployments where naive approaches break down. Similarly, the consistent handover reduction (66.7–75.0%) across all satellite densities confirms that Demand-Aware will continue to perform well as LEO constellations grow denser. This combination of strong baseline performance and excellent scaling behavior positions our framework as a robust solution for both current and future SAGIN deployments.

Several practical considerations arise when contemplating real-world deployment. For Algorithm 1, the assumption of perfect channel state information should be relaxed in practice. Actual implementations would need to estimate channel gains from pilot signals and incorporate prediction errors into the optimization. The single-visit constraint could also be relaxed to allow multiple UAV passes over the coverage area, trading increased mission time for better energy efficiency. For Algorithm 2, the fixed demand threshold could be replaced with an adaptive mechanism that adjusts based on observed channel conditions and remaining mission time. The algorithm could also be extended to coordinate handovers across multiple UAVs to avoid simultaneous switching events that might overwhelm satellite resources.

Despite these potential enhancements, the current framework already demonstrates strong practical viability. The algorithms require no specialized hardware beyond standard UAV platforms and commodity satellite transceivers. The computational requirements are modest enough for embedded flight computers. The performance improvements are substantial enough to justify implementation effort. Most importantly, the framework is modular: the two stages can be deployed independently, allowing operators to adopt whichever component best addresses their primary bottleneck.

## 6. Conclusions

In this paper, we address the data collection and offloading problem by integrating UAVs trajectories planning and LEO satellite selection in SAGIN. The problem is formulated to minimize the energy consumption with the constraints on multi-dimension variables.

Specifically, in the data collection phase from IoT to UAV, the algorithm is designed to optimize the IoT pairing, power optimization, UAV trajectory planning. In the data offloading phase from UAV to LEO, a real-time LEO satellite selection mechanism joint with STK is proposed. Finally, simulation results verified the effectiveness of the proposed approach, with about 10% less energy consumption compared with the benchmark algorithm. oilk,

## References

1. Jia, Z.; Sheng, M.; Li, J.; Niyato, D.; Han, Z. LEO-satellite-assisted UAV: Joint trajectory and data collection for Internet of Remote Things in 6G aerial access networks. *IEEE Internet Things J.* **2020**, *8*, 9814–9826.
2. Xiao, Y.; Ye, Z.; Wu, M.; Li, H.; Xiao, M.; Alouini, M.-S.; Al-Hourani, A.; Cioni, S. Space-air-ground integrated wireless networks for 6G: Basics, key technologies and future trends. *IEEE J. Sel. Areas Commun.* **2024**.
3. Duan, S.; Wang, D.; Ren, J.; Lyu, F.; Zhang, Y.; Wu, H.; Shen, X. Distributed artificial intelligence empowered by end-edge-cloud computing: A survey. *IEEE Commun. Surv. Tutor.* **2022**, *25*, 591–624.
4. Wei, Q.; Chen, Y.; Jia, Z.; Bai, W.; Pei, T.; Wu, Q. Energy-efficient caching and user selection for resource-limited SAGINs in emergency communications. *IEEE Trans. Commun.* **2024**.
5. Pan, G.; Ye, J.; An, J.; Alouini, M.-S. Latency versus reliability in LEO mega-constellations: Terrestrial, aerial, or space relay? *IEEE Trans. Mob. Comput.* **2022**, *22*, 5330–5345.
6. Jia, Z.; Cui, C.; Dong, C.; Wu, Q.; Ling, Z.; Niyato, D.; Han, Z. Distributionally robust optimization for aerial multi-access edge computing via cooperation of UAVs and HAPs. *IEEE Trans. Mob. Comput.* **2025**.
7. Mao, S.; He, S.; Wu, J. Joint UAV position optimization and resource scheduling in space-air-ground integrated networks with mixed cloud-edge computing. *IEEE Syst. J.* **2020**, *15*, 3992–4002.
8. Zhao, C.; Liu, J.; Sheng, M.; Teng, W.; Zheng, Y.; Li, J. Multi-UAV trajectory planning for energy-efficient content coverage: A decentralized learning-based approach. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 3193–3207.
9. Mozaffari, M.; Saad, W.; Bennis, M.; Nam, Y.-H.; Debbah, M. A tutorial on UAVs for wireless networks: Applications, challenges, and open problems. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2334–2360.
10. Tao, Y.; Liu, L.; Liu, S.; Zhang, Z. A survey: Several technologies of non-orthogonal transmission for 5G. *China Commun.* **2015**, *12*, 1–15.
11. Fang, X.; Feng, W.; Wang, Y.; Chen, Y.; Ge, N.; Ding, Z.; Zhu, H. NOMA-based hybrid satellite-UAV-terrestrial networks for 6G maritime coverage. *IEEE Trans. Wireless Commun.* **2022**, *22*, 138–152.
12. Jia, Z.; Cao, Y.; He, L.; Wu, Q.; Zhu, Q.; Niyato, D.; Han, Z. Service function chain dynamic scheduling in space-air-ground integrated networks. *IEEE Trans. Veh. Technol.* **2025**.
13. Huang, C.; Chen, G.; Xiao, P.; Xiao, Y.; Han, Z.; Chambers, J. A. Joint offloading and resource allocation for hybrid cloud and edge computing in SAGINs: A decision assisted hybrid action space deep reinforcement learning approach. *IEEE J. Sel. Areas Commun.* **2024**, *42*, 1029–1043.
14. Jia, H.; Wang, Y.; Wu, W. Dynamic resource allocation for remote IoT data collection in SAGIN. *IEEE Internet Things J.* **2024**, *11*, 20575–20589.
15. Seyedi, Y.; Rahimi, F. A trace-time framework for prediction of elevation angle over land mobile LEO satellites networks. *Wirel. Pers. Commun.* **2012**, *62*, 793–804.
16. d O Costa, P. R.; Rhuggenaath, J.; Zhang, Y.; Akcay, A. Learning 2-opt heuristics for the traveling salesman problem via deep reinforcement learning. *Asian Conf. Mach. Learn.* **2020**, 465–480.