



OPEN ACCESS

EDITED BY

Athirah Mohd Ramly,
University of East London, United Kingdom

REVIEWED BY

Ahmad Anwar Zainuddin,
International Islamic University Malaysia,
Malaysia
Khairayu Badron,
International Islamic University Malaysia,
Malaysia

*CORRESPONDENCE

Hitesh Mohapatra,
✉ hiteshmohapatra@gmail.com

RECEIVED 23 August 2025

ACCEPTED 06 October 2025

PUBLISHED 29 October 2025

CITATION

Mohapatra H (2025) Adaptive ant colony
methods for UAV LEO coordination in non
terrestrial IoT.

Front. Commun. Netw. 6:1691346.
doi: 10.3389/frcmn.2025.1691346

COPYRIGHT

© 2025 Mohapatra. This is an open-access
article distributed under the terms of the
[Creative Commons Attribution License \(CC BY\)](#).
The use, distribution or reproduction in other
forums is permitted, provided the original
author(s) and the copyright owner(s) are
credited and that the original publication in this
journal is cited, in accordance with accepted
academic practice. No use, distribution or
reproduction is permitted which does not
comply with these terms.

Adaptive ant colony methods for UAV LEO coordination in non terrestrial IoT

Hitesh Mohapatra*

School of Computer Engineering, Kalinga Institute of Industrial Technology (KIIT) Deemed to be
University, Bhubaneswar, Odisha, India

Introduction: This work presents an adaptive ant colony (AdCO) framework for dynamic task management in heterogeneous Non-Terrestrial Network–Internet of Things (NTN-IoT) systems integrating Unmanned Aerial Vehicles (UAVs) and Low Earth Orbit (LEO) satellites. The framework addresses key challenges such as stochastic mobility, intermittent connectivity, and latency-sensitive operations common in large-scale IoT deployments.

Methods: The proposed approach employs adaptive pheromone learning, heuristic control, and multi-timescale scheduling. It follows a hierarchical co-optimization strategy, where UAV swarms perform edge-side task allocation while LEO satellites handle relay scheduling during orbital passes. Event-triggered pheromone resets and distributionally robust cost modeling are introduced to maintain stability and adaptability under dynamic network conditions.

Results: Simulation results demonstrate superior performance compared to classical Ant Colony Optimization (ACO) and recent meta-heuristic methods. The proposed model achieves higher task completion ratios, reduced end-to-end latency, and enhanced energy-normalized throughput across different orbital configurations, traffic patterns, and link failures.

Discussion: The findings confirm the efficiency and resilience of the proposed framework in NTN-IoT operations. Its adaptability makes it suitable for critical applications such as disaster response, precision agriculture, and maritime monitoring, where real-time coordination and reliability are essential.

KEYWORDS

non-terrestrial IoT, UAV swarms, LEO satellites, adaptive colony optimization, dynamic task allocation, scheduling, edge computing, connectivity-aware orchestration

1 Introduction

Non-terrestrial IoT (NTN) enhances the terrestrial based internet of things (IoT) using unmanned aerial vehicle (UAV) swarms and LEO satellite constellations. It is showing growing potential as a scalable technology for wide-area IoT applications. NTN provides an opportunity for on-demand sensing, *in-situ* edge computing and beyond-line-of-sight data relaying in areas with such limited or disrupted terrestrial infrastructure (Kahraman et al., 2023). However, several challenges exist. Dynamic link topology, time-varying link budgets, fragmented orbital pass windows, and the trade-offs between energy, latency, and reliability render global coordinated tasking across multiple timescales a challenging optimization problem. Classical swarm intelligence techniques like ant colony optimization (ACO) can cope with static or slowly changing routing problems. However, the fixed parameterization and the narrow detection range of changes usually leads to pheromone stagnation or late re-

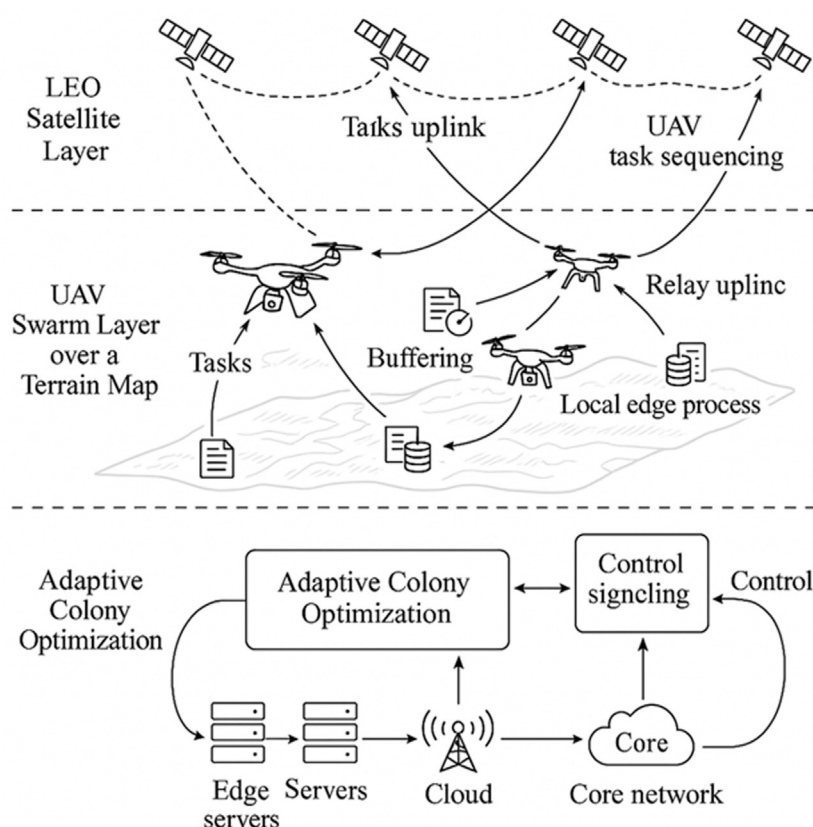


FIGURE 1
System overview of UAV-LEO-ground segments, data flows, scheduling layers, control loop.

planning when the environment changes suddenly (Deng et al., 2022). To address this issue, an adaptive colony (AdCO) optimization model is presented to tackle these shortcomings. The work has been compared with three kinds of coordinated approaches: (i) standard ACO with fixed powers and evaporation, (ii) bounded-pheromone variants (such as MAX-MIN ACO for better control of exploitation), and (iii) a policy heuristic scheduler that combines deadline-aware prioritization and proximity-based assignment (Zhang et al., 2025). This analysis decouples the impact of the event-triggered adaptation, the connectivity-aware heuristics, and the robust cost shaping on timeliness, energy efficiency, and tail-latency robustness for realistic UAV and low Earth orbit (LEO) dynamics.

Combining LEO satellite constellations with UAV Swarm provides a solution to resilient and large area IoT service. Such integration is needed in territories where the terrestrial infrastructure is not possible, or is not reliable, or is too costly (Mai et al., 2023). Disaster-stricken areas, marine corridors, farming fields and isolated industry are all examples of such settings. In such scenarios, sensing tasks are spatio-temporally distributed and bursty. Opportunities for connectivity are subject to orbital visibility, the condition of the atmosphere and availability of spectrum (Jubair et al., 2025). UAVs provide mobility and fast deployment as well as edge computing, but are subject to limitations in energy storage, flight dynamics, and onboard computing power. On one hand, LEOs satellites are able to provide wide-area relays

and backhaul services, while on the other hand, they have a sporadic contact window, the ability to induce velocity Doppler variations and competition for bandwidth with traffic classes. Controlling both the tasks' execution and the data delivery over UAV and LEO layers is achieved by managing two synchronized dynamical processes (Zhu et al., 2023). The first refers to spatiotemporal task allocation and motion planning in UAV network. The second such is contact-aware relay scheduling that adapts with the varying LEO pass structure. Figure 1 describes the system architecture that includes UAVs, LEOs and ground segments, the data flows, the scheduling layers and the control loop.

In this system, the decision space is combinatorial and non-stationary. Estimates of task priority change with time, shifting according to deadlines and arrivals. Quality varies with elevation angles (EAs) and mobility directions. The buffer occupancy, the energy level, and the computational state all vary on different time scales as well. There have been mostly two types of approaches to managing these interacting dynamics: classical (suboptimal) techniques, including static task assignment, single-timescale routing, and topology-agnostic heuristics, and optimal solutions with simplified and static dynamics. They often lead to a tradeoff between high re-planning overhead, and performance loss due to out-of-date policies (Li et al., 2025b). The operational goals are multi-criteria in the sense. These objectives involve low end-to-end latency (i.e., long life time regardless of the number of deadlines missed), low deadline misses, strict energy consumption, and link

quality in terms of uncertain channel conditions and outages (Hasanain et al., 2025). Partial observability further complicates the problem. Ephemeris predictions or channel estimates can be frequently unreliable. Sudden changes in the working environment (e.g., sudden task bursts or unexpected satellite pass losses) also account for another challenge (Peng et al., 2025). So, an efficient coordinator must be able to provide a fast decision-making on the fly and be adaptive and robust. It has to support immediate reconfiguration and be free of local oscillations such that low performance is not realized in very dynamic and resource-constrained NTN-IoT scenarios with very low end-to-end latency sensitivity (e.g., UAV deployment) (Behjati et al., 2025).

1.1 Research gap and proposed contribution

Existing work falls into three key categories to leverage UAV satellite communications for IoT coordination, but this existing work suffers from the challenges of fast changing NTN conditions (Gu et al., 2023). The first one is of classical swarm-based optimizers like fixed-parameter ACO and bounded-pheromone variants. Such methods work well for combinatorial search in static or slowly varying routing and task allocation. But they have poor adaptability to the scripting topology changing and sudden contact loss and bursty task arrival time. Pheromones stagnate and do not recover in time when operation conditions experience a large change (Lozano-Cuadra et al., 2025). The second type consists of heuristic schedulers which mix deadline-aware rules with location-oriented or load-balancing heuristics. These methods have low computation complexity, however, these methods do not have any capability to predict the link variation, led by the visibility. As a consequence, their performance deteriorates when LEO contact opportunities or channel conditions are not as good as expected (Lozano-Cuadra and Soret, 2024). The third type is learning-based methods, such as the reinforcement learning. These methods are adaptive under non-stationary environments but require sufficient amounts of training data and are not capable of dealing with the distribution shift between different orbits and terrains. They could also be inadequate to satisfy real-time decision making criteria in the absence of watching for sample efficiency and safety guarantees (Maric et al., 2025). However, three important gaps persist. First, it is relatively lacking in the cross-timescale coordination support to optimize UAV task scheduling and LEO relay scheduling. Second, tail risks and outages are not fully taken into account in the objective formulation in the current methods. Third, the majority of existing algorithms do not include events mechanisms, which keep the population from stagnation while letting the system converge in fast changing environments (Panaitopol et al., 2023).

In this respect, the proposed AdCO is designed to fill this gap by introducing three main differences as introduced below with respect to existing methods. The latter integrates a fast loop for sequencing the UAV tasks, and a slower loop to solve the LEO relay scheduling problem. This temporal alignment adjusts the decision making rate with the physical dynamics of the UAV mobility and satellite pass windows (Yosri et al., 2024). Secondly, it uses event-triggered partial pheromone resets using online drift detection. This mechanism allows to explore again in a controlled way in case of contact geometry or of traffic patterns change and at the same time to retain useful historical

information. Third, it integrates distributionally robust cost shaping with risk-sensitive penalties. These punishments encourage meeting deadlines instead of reliability in unreliable links, which ultimately reduces tail latency and failures due to outages rather than serving the mean (Liu et al., 2024). These features combined return a coordinator, which is aware of the connectivity, risk-sensitive, in addition to being able to remain responsive. Experimental validation compares AdCO against three known baselines: fixed-parameter ACO, MAX-MIN ACO, and an heuristic scheduler with a combination of deadlines and proximity. The numerical findings quantitatively justify the advantages of multi-time scale learnability, event-triggered restoration, and robust objective shaping on the fly in dynamic NTN-IoT regimes (Bazrafkan et al., 2025).

This study is conducted entirely within a controlled simulation environment by setting using standards and repeatable components. Orbit dynamics and visibility windows are calculated based on publicly accessible Two-Line Element (TLE) datasets. These are propagated using an Simplified General Perturbations 4 (SGP4) model to produce time-stamped elevation and Doppler and line-of-sight contact profiles over the target region (Kohani and Zong, 2019). The mobility of UAVs is expressed by a 3D kinematic model, which takes into account energy-aware propulsion curves. These are the task fields, obtained from spatial point processes with intensities which are configurable and can be either clumped or uniform. The model has the flexibility to tune the tasks' arrival rate, deadlines, data size and computational workload (Zhao et al., 2021). The communications layer utilizes a path loss model in the presence of elevation-dependent path with log-normal shadowing and stochastic outage model. This setup simulates UAV-LEO uplinks and satellite ground downlinks. Furthermore, capacity limitations and sharing are modelled with rate envelopes that depend on time.

The paper is organized into seven sections to maintain clarity and cohesion. Section 1 introduces the motivation for integrating UAV swarms with LEO satellites, emphasizing coordination challenges and research gaps addressed by the adaptive colony optimization framework. Section 2 reviews related work on UAV task allocation, integrated networks, colony optimization strategies, heuristic schedulers, and learning-based coordinators, while highlighting their limitations under dynamic NTN conditions. Section 3 presents the system model and problem formulation, defining entities, visibility and capacity constraints, mobility and energy models, traffic parameters, decision variables, and the distributionally robust objective. Section 4 describes the proposed method, including solution encoding, connectivity-aware heuristics, adaptive transition rules, event-triggered pheromone resets, and the bi-level scheduling approach. Section 5 details the simulation environment, experimental setup, baseline methods, evaluation metrics, and reproducibility settings. It also reports results and discussion, comparing the proposed method with representative coordinators and presenting ablation studies under outages, workload bursts, and contact variability. Finally, Section 6 concludes with key findings, practical implications, limitations, and directions for future research.

2 Literature review

Early research on UAV swarm task allocation primarily relied on centralized optimization methods such as mixed-integer linear

programming, Hungarian algorithms, and auction-based mechanisms (Pham et al., 2020). Traditional approaches, including the Consensus-Based Bundle Algorithm and its variants, performed well in static settings with predetermined task sets and known agent capabilities. Deterministic methods offered mathematically guaranteed convergence and optimality for problems with fixed constraints (Wang et al., 2024). Hungarian-based strategies, such as the Decentralized Hungarian Algorithm, further ensured efficient polynomial-time complexity for bipartite matching structures (Li et al., 2025a). However, these classical techniques face major challenges in dynamic and non-terrestrial environments. Their static parameterization makes them ill-suited for handling time-varying topologies, intermittent connectivity, and stochastic task arrivals (Chen et al., 2018). Moreover, computational costs grow rapidly with swarm size and task complexity, limiting real-time applicability. Centralized designs also introduce single points of failure and demand extensive inter-agent communication, which is impractical in bandwidth-constrained UAV-LEO satellite operations.

Drawbacks of classical optimization techniques have encouraged the utilization of metaheuristic heuristic methods for UAV task assignment also. It was established that ACO, Particle Swarm Optimization (PSO), and Genetic Algorithms (GA) were the mostly used approaches in the existing study (Divband Soorati et al., 2019). Where traditional ACO implementations have met with considerable success in solving combinatorial routing problems using pheromone-based communication and distributed decision making to arrive at near-optimal solutions (Bonyadi et al., 2014). In the same vein, PSO variants including multi-objective and hybrid models were effective in handling conflicting demands such as energy efficiency and deadline satisfaction in Zhu et al. (2024). These swarm intelligence approaches are by nature adapted to parallelism, scalability and robustness against the failure of individual members. Their search strategy in a probabilistic framework encourages exploration of various solution spaces, which gradients methodologies tend to fall into local minima. However, classic metaheuristics are still characterized by static parameterization, which restrains versatility in dynamic operational conditions (Zhang et al., 2024). These fixed rates often deteriorate their performance under changes, for they may find themselves stuck in prematurely converged or stagnant state. In addition, the absence of adaptation strategies considering the environment prevents these solutions from dealing with regime shifts, orbital pass variability, and non-stationary traffic demands. Among the latest ones, parameter-adaptive ACO with dynamic hybrid mechanisms attempts to largely overcome this issue but still acts more reactively than predictively (Pirandola, 2021).

Recent investigations focus on the increasing use of machine learning in realizing enhanced UAV swarm coordination. This is where the inhomogeneous reinforcement learning comes in. The multi agent reinforcement learning provides the means for UAVs to learn the collaboration behavior by the trial and error interactions. These mechanisms permit adjustments to be made to changing environmental conditions without reprogramming (De Forges de Parny et al., 2023). Recent works utilizing Deep-Q Networks and Actor Critic based algorithms have achieved competitive performance in rescue and target tracking, and area coverage missions. Methods based on learning work well in environments where only a partial view of the world is available, in non stationary

conditions, and with complex multi agent interactions. They are ideal for long-term missions with changing objectives as they have the capacity of continuous learning and adaptation (Miller, 2025). Reinforcement learning libraries could also support more complex state representations, where graph like observations and attention mechanisms are used to represent inter agent relations and dynamic interactions with the environment. These methods, however, are confronted with sample efficiency issues. They need large amount of training data, which is scarce in safety-critical applications (Zengshan et al., 2025). Due to black-box policies, verification and certification used in aerospace side are challenged also. Convergence guarantees are often weak or non-existent and initial deployment may involve sub-optimal or unsafe behaviour during exploration (Soret et al., 2024).

Distributed optimization methods have been proposed as an intermediate option between the centralized control and the full autonomy. These techniques use consensus algorithms, game theory, distributed constraints optimization, for the management of the coordination of the UAV swarm (Melillos et al., 2019). The Distributed Greedy Bundles Algorithm is a substantial advance for submodular maximization for MAS. It has formal and approximation guarantees while working in polynomial time (Wang et al., 2021). Game theoretic models unify UAV interactions as two basic or general games in order to search for the Nash equilibrium or generalised Nash equilibrium with resource constraints. Event-activated communication schemes reduce the overall usage of bandwidth, while maintain convergence properties and this is useful in UAV to satellite networks with intermittent connectivity (Li C. et al., 2021). These decentralised methods provide formal convergence guarantees, handle agent heterogeneity well, and scale well to swarm size. Moreover, the decentralized nature of these SLAMs make them more capable of handling communication failures and agent drops, both of which are prevalent in aerial agent operation. However, methods relying on consensus often converge slowly due to the needs to have multiple rounds of communication to reach an agreement (Li Y. et al., 2021). Game theoretic solutions can lead to inefficient equilibria, in particular in adversarial settings. Furthermore, the rationality assumption about the agents behavior may not be satisfied due to hardware faults, communication delays or adversarial environments.

The development of the uncertainty effects of UAV satellite scenarios has resulted in growing efforts to address robust optimization and distributionally robust optimization (Stodola et al., 2025). Polyhedral convex sets are used to describe parameter uncertainties in uncertain resource allocation frameworks. Such frameworks take care of errors due to channel estimation, inaccuracy in dynamic prediction and changes in traffic flow. Distributionally robust optimization methods use ambiguity sets and coherent risk measures like Conditional Value at Risk to obtain performance bounds that are robust to shifts in the distribution (Fakoor et al., 2016). These methods directly naturally include the model mismatches and provide worst case performance outcomes and enhanced tail risk management. By including uncertainty quantification we can make trade-off between nominal performance and robustness. Distributionally robust methods mitigate the conservatism of bounded uncertainty sets while still being computationally tractable (Wei Jiang et al., 2024). However, strong optimization generally yields conservative solutions that favor the worst case protection, while sacrificing the average case performance. The quality of solutions is sensitive to the

TABLE 1 Comparative analysis of coordination methods for UAV-LEO NTN systems.

Method	Opt.	Adapt.	Scale.	RT	Rob.	Comp.	NTN
Classical (MILP/Hungarian)	H	L	L	L	L	H	L
Fixed ACO	M	L	M+	M	M	M	M
PSO	M	L+	M	M	M	M	M
Multi-Agent RL	L+	H	M	L	M+	H	M+
Distributed Consensus	M	M	H	M	H	M	H
Robust Optimization	M	L	M	L+	H	H	M+
Multi-Timescale Hier	M+	M	H	M+	M	M+	H
Hybrid Meta-heuristic	M	M+	M	M	M+	H	H
Event-Triggered Dynamic	M	H	M+	H	M+	M	H
Proposed AdCO	H	H	H	H	H	M	H

Legend: H, high; M, medium; L, low, + indicates upper bound of category. The bold line represents the efficiency of the proposed AdCO model as comparative analysis against existing methods.

choice of uncertainty or ambiguity sets, which do not always capture actual operational situations. They require a more complex processing that demands a very high amount of computational power and thus they are not suitable for real time application. Moreover, conservative solutions potentially can be over-conservative, when uncertainty estimates are not accurate or when systems are functioning properly for long periods of time (Basil et al., 2025).

In recent work, the multi timescale nature of UAV satellite coordination is emphasised, as fast UAV dynamics and slower orbital mechanics imply having different planning horizons (Yang et al., 2024). Two-level optimization methods optimize temporal scales away from short term resource allocation (down) and long term trajectory planning (up), thus making it possible to decompose complicated coordination tasks effectively. Hierarchical structures break the entire mission task into several layers, such as high level mission planning, medium level task assignment, and low level trajectory implementation. Each layer acts on multiple time scales (Zhou and Zheng, 2013). Multi objective optimization paradigms handle trade off among conflicting objectives, such as short term power consumption and long term mission success. These techniques match algorithmic effort with the dynamic of the system, to decrease computation cost by maintaining solution quality (Toppen et al., 2017). The layered decomposition enables specialized algorithm design for multiple timescales and easy integration into legacy control systems. However, suboptimal solutions can be obtained with hierarchical strategies as a result of decomposition approximations. Flow of information loss between the layers may lead to the inconsistency in the decision making and compromise the overall performance Han et al. (2024). Coordinating over multiple time scales must also be mindful of what the interface should look like and possibly add another level of communication between machines.

Contemporary research increasingly focuses on hybrid approaches that combine classical optimization, metaheuristics, and machine learning (Li et al., 2024). Hybrid metaheuristic algorithms integrate multiple optimization techniques, such as merging ant colony optimization with local search or combining genetic algorithms with particle swarm optimization. Learning augmented optimization applies

machine learning to guide traditional optimizers by predicting good initial solutions or dynamically adjusting parameters. Reinforcement learning frameworks incorporate classical control strategies to ensure safety while enabling adaptive behavior learning. These hybrid methods exploit the complementary strengths of different techniques, often achieving better performance than individual approaches (Liu et al., 2022). Classical components offer convergence guarantees and optimal solutions for well defined subproblems, while learning components enhance adaptability and handle uncertainties. The modular architecture supports testing, verification, and incremental deployment. However, hybrid systems increase complexity in design, implementation, and maintenance (Wang et al., 2022). Interactions between components can cause unexpected behaviors or performance degradation. Integration challenges also arise from differences in computational requirements, convergence speeds, and solution representations across hybrid modules.

The dynamic nature of UAV satellite environments requires real time adaptation, leading to advances in dynamic task scheduling and event triggered optimization. Dynamic priority mechanisms allow task importance to change based on environmental conditions, mission progress, or external events (Paek et al., 2019). Event triggered algorithms initiate optimization only when significant changes occur, reducing computational costs while maintaining responsiveness. Online learning methods continuously update models and policies using observed system behavior and performance feedback (He et al., 2024). Adaptive resource allocation techniques adjust to varying system conditions, load fluctuations, and constraint modifications. These dynamic strategies enable responsive adaptation while preserving computational efficiency (Wang et al., 2023). Event triggered mechanisms reduce communication and computation demands compared to continuous optimization. Online learning supports gradual performance improvements and adaptation to long term system changes. However, event triggered systems require careful selection of triggering conditions to maintain stability. Excessively rapid adaptation can cause oscillatory behavior or instability in noisy or transient environments (Zhou et al., 2022). Online learning methods may also face issues such as catastrophic forgetting and slow convergence under rapidly changing conditions.

Table 1 presents trade offs on UAV-LEO coordination mechanisms and the importance of balanced performance. Traditional optimization approaches, such as MILP and Hungarian algorithms, achieve strong optimality but are not adaptive, scalable, and computationally efficient in real time, which hinders the applications in dynamic environments. Classical ACO and PSO provide moderate results and miss the generality of the approach when the environment is dynamic since parameters are constant. Machine learning especially multi agent reinforcement learning exhibit high flexibility and robustness at the cost of high computational complexity, and weaker optimality guarantees. Distributed consensus and robustified optimization T Better scalability and uncertainty handling but worse reactivity and optimality. No balance is reached by hybrid metaheuristics and multi timescale frameworks as these have high scalability and are NTN suitable, while at the same time they have adaptability and optimality constraints. Event-Led Scheduling is high achieving in real time but compromises predictive accuracy. The resulting adaptive colony optimization model integrates event triggered adaptability, robust cost shaping and connectivity aware heuristics, and reaches a trade-off between solution quality and computational complexity in dynamic UAV-LEO coordination.

3 Proposed work

The proposed AdCO framework overcomes the primary shortcomings of current UAV and LEO coordination schemes. It combines adaptation, robust optimization and multi-timescale coordination in one framework. Traditional optimization methods are often notoriously plagued by static parameterization and low real-time efficiency. On the other hand, the AdCO uses event-triggered parameters update and connectivity-based heuristics. Such techniques make fast responses to topology changes, orbital pass fluctuation and traffic non-stationarity with convergence guarantees. The framework addresses the slow adaptation and pheromone stagnation of fixed-parameter metaheuristics. It does so by dynamically modulating exploration and exploitation by means of feedback-generated learning coefficients and drift-sensitive pheromone partial resets. This design keeps all the time an exploration according to changing operating conditions. AdCO combats the sample inefficiency and black-box nature of RL methods with domain-specific heuristics inspired by orbital mechanics and communication theory. This has the benefit of making decisions more interpretable, and reduces the training burden. A bi-level scheduling structure is adopted to answer the problem of the scalability issues in consensus based methods. Fast UAV task sequence is decoupled with slow LEO relay coordination, making it very low communication overhead while still global coordination. The proposed framework also mitigates the conservatism of robust optimization with distributionally robust cost shaping. This method guards against the truly disastrous risks, but does not become too pessimistic. Last but not least, AdCO alleviates the drawbacks with event-triggered scheduling. This includes predictive visibility forecasting and proactive pheromone structuring ensuring that the framework may predict

even connectivity windows and not only reacting to them. Algorithm 1 summarizes the overall procedure of the AdCO algorithm. Figure A1 illustrates the flowchart of the proposed model.

Input: Tasks $\mathcal{K}$, UAVs $\mathcal{U}$, satellites $\mathcal{S}$, time horizon $t = 1:T$; visibility $V_{u,s}(t)$, rates $R_{u,s}(t)$, loss $p_{u,s}(t)$; slot Δt ; weights $(\alpha, \beta, \gamma, \delta)$; AdCO params $(\tau_0, \rho, Q, \zeta, \lambda, \alpha_{\text{CVar}}, \lambda_\alpha, \lambda_\beta, \phi_{\text{elite}}, K, \theta, M)$.

Output: UAV task sequences $\{\pi_u\}$ and relay fractions γ

1 Init: Set pheromones $\tau_{i,j} \leftarrow \tau_0$, $\alpha(1), \beta(1)$; initialize $\bar{R}_{u,s}(t)$, drift stats $D(t)$.

2 for epoch $e = 1, 2, \dots$ **do**

3 Build relay graph $\mathcal{G}_{\text{relay}}$; update CVaR stats.

4 for slot $t = (e-1)K + 1$ **to** eK **do**

// Drift Handling

5 Compute $D(t)$; **if** $D(t) > \theta$ **then**

reset $\tau_{i,j} \leftarrow (1 - \zeta)\tau_{i,j} + \zeta\tau_0$

7 end

// Ant Solution Construction

8 for $r = 1$ **to** M **do**

foreach $u \in \mathcal{U}$ **do**

$\pi_u^{(r)} \leftarrow \emptyset$; **while** tasks remain **do**

Compute heuristic $\eta_{i \rightarrow j}(t)$; sample j via $P_{i \rightarrow j}^{(u)} \propto (\tau_{i,j})^\alpha \eta_{i \rightarrow j}^\beta$; append j

end

end

Allocate relays: $y_{u,s}^{(r)}(t) \leq V_{u,s}(t)$, $\sum_k a_{u,k}^{(r)} y_{u,s}^{(r)} d_k \leq R_{u,s} \Delta t$.

Repair infeasible schedules: enforce CPU/energy limits.

16 end

// Candidate Evaluation

17 for $r = 1$ **to** M **do**

Compute L_k , E_u^{tot} , reliability, CVaR; objective:

$$J_r = \sum_k (\alpha L_k + \beta \mathbf{1}_{\{L_k > \tau_k\}}) + \sum_u \gamma E_u^{\text{tot}} - \delta \sum_k \log P(\text{succ}_k) + \lambda \text{CVar}.$$

19 end

// Pheromone Update

20 Select elite set $\mathcal{E}_t$; **foreach** edge (i, j) **do**

$\Delta \tau_{i,j} = \sum_{r \in \mathcal{E}_t} \frac{Q}{J_r + \lambda \text{CVar}}$;

$\tau_{i,j} \leftarrow (1 - \rho)\tau_{i,j} + \Delta \tau_{i,j}$

23 end

// Adapt α, β

24 Update α, β via regret and QoS indicators.

25 end

26 end

27 return best $\{\pi_u\}$ and relay fractions γ

Algorithm 1. AdCO: Adaptive Colony Optimization for Dynamic UAV-LEO Coordination

1 The simplified version of the proposed algorithm is available under Appendix Section.

TABLE 2 Simulation environment specifications and data sources.

Component	Tools/Methods	Data sources	Configuration
Orbital Dynamics	SGP4 Propagator with TLE Data	CelesTrak TLE Sets, NORAD Elements	Walker Constellation Subset, Pass Prediction
UAV Mobility	3D Kinematic Model with Energy Constraints	Waypoint Sequences, Speed/Acceleration Profiles	Max Velocity, Energy Capacity, Compute Rate
Communication Channel	Elevation-dependent Pathloss + Log-normal Shadowing	ITU-R P.619 Models, Measured Link Budgets	Frequency Band, Antenna Gains, Noise Floor
Task Generation	Spatial Point Processes (Clustered/Uniform)	Synthetic Hotspot Distributions, Real Coverage Maps	Deadline Constraints, Data Volume, Compute Load
Traffic Modeling	Poisson and ON-OFF Burst Processes	Mission-specific Arrival Patterns	Arrival Rate, Burst Duration, Correlation
Network Topology	Dynamic Graph with Time-varying Adjacency	Graph Theory Adjacency Matrices	Contact Windows, Visibility Matrices
Energy Model	Propulsion + Compute + Communication Components	Battery Discharge Models, Motor Efficiency Curves	Power Models for Flight/Compute/Radio
Performance Metrics	Completion Ratio, Latency CDF, Energy per Task	Time Series Data, Statistical Distributions	Percentiles, Confidence Levels, Sample Size
Statistical Analysis	NumPy/SciPy with Bootstrap Confidence Intervals	Monte Carlo Sampling, Hypothesis Testing	Seed Control, Reproducibility Parameters

3.1 Simulation environment

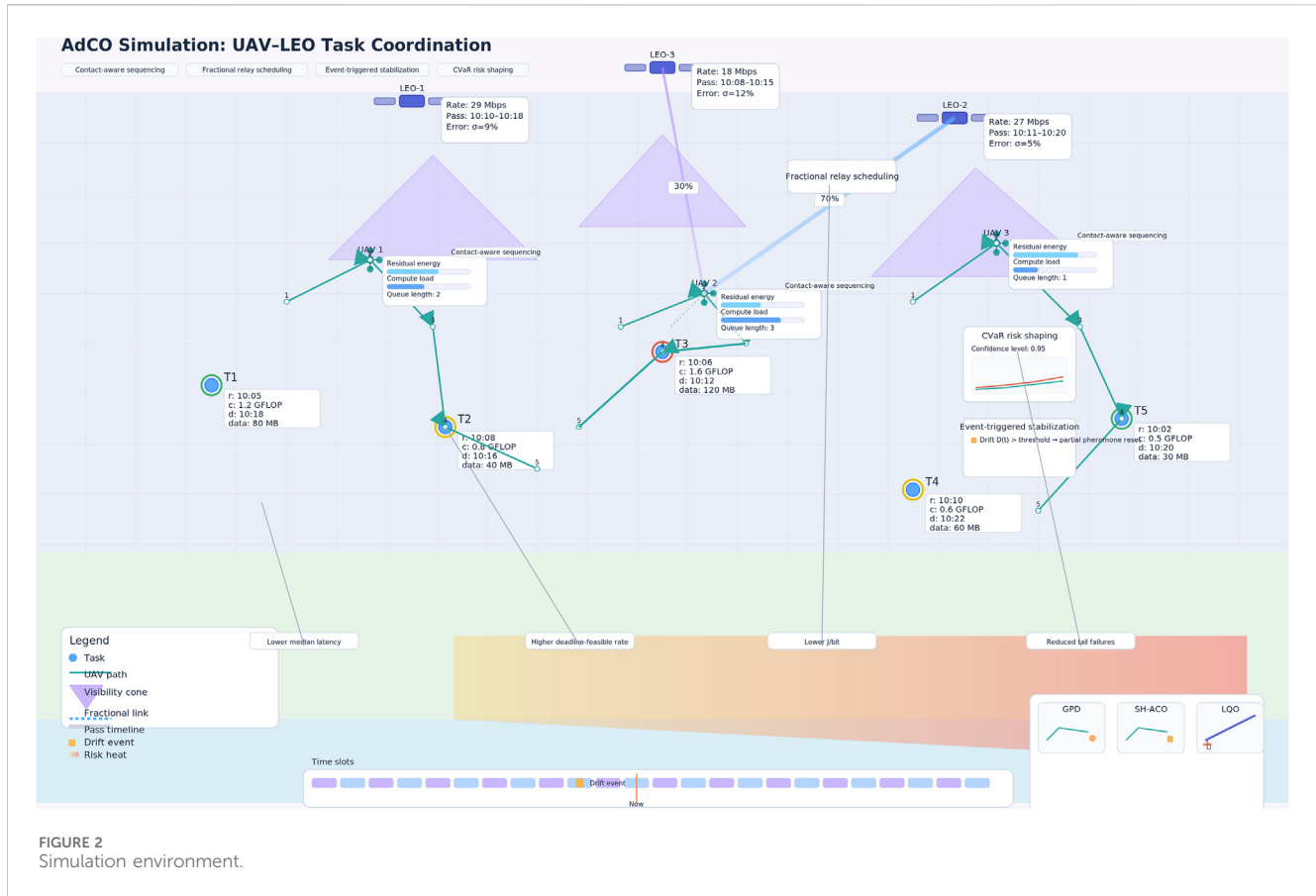
A discrete event simulation environment written in Python is used in the evaluation framework. It uses accepted models and benchmark datasets to guarantee the repeatability and the reality of the UAV and LEO coordination algorithms assessment. The orbital mechanics layer is based on TLEs supplied by NORAD and CelesTrak. These TLEs are propagated using the SGP4 algorithm via the Python SGP4 library. This procedure creates time-tagged time visibility windows, elevation profiles and Doppler shift patterns for the specific region under consideration. The mobility of the UAVs is represented with 3D kinematic equations dedicated to energy-aware flight dynamics. The model combines physical realistic propulsion power demand curves, maximum velocity limits, and battery discharge laws grounded on commercial UAV specifications. The communications system is based on ITU R P.619 recommended elevation-dependent path loss models. It also includes log-normal shadowing with $\sigma = 4$ dB and stochastic outage processes. These extended generate realistic link quality fluctuations due to both atmospheric and geometric conditions.

Task distribution applies spatial point processes, such as clustered hotspot distributions and uniformly random placement. Arrival rates are tunable and are modeled after Poisson and ON-OFF bursty patterns. These task setups are based on realistic mission settings such as disaster response and agricultural monitoring. The network structure is changing as a dynamic graph by time-varying adjacency matrices. They capture UAV-LEO contact windows, which are intermittent, ground station visibility, and inter-UAV communication slots. The communication is limited due to range and line of sight. The energy model is for both the aircraft's power for forward and turning flight as well as the computing power for onboard processing and communication power for the data packets traveling between ground and air. This allows for a more

accurate assessment of trade-offs in energy efficiency. Performance statistics are calculated using statistical analysis on the NumPy and SciPy libraries. Bootstrap confidence intervals and Monte Carlo sampling are implemented over 50–100 independent random seeds. These approaches ensure the statistical significance and reproducibility of the results. Table 2 the basic simulation parameters and the assigned ranges are reported. It is defined in terms of the region size, the size of the swarm and the subset of the satellite constellation. Workload characteristics are determined by task arrival rates and deadline typologies. The communication environment is modeled in terms of height thresholds and shadowing factors. The limited energy resource of UAV is characterized by battery capacity ranges. Table 2 additionally, summarizes the number of trees used, the time in which each virtual experiment is executed and the number of Monte Carlo runs. These criteria provide a structured approach to systematically comparing algorithm performance across a range of operational conditions. Figure 2 illustrates the simulation environment.

4 Problem formulation

The problem formulation provides a structured platform where the mathematical depiction of the system entities, performance indices and the operational constraints helps in synthesizing the AdCO algorithm. Table 3 presents the notations used. Binary and continuous decision variables are adopted in the formulation to model task allocation and relay scheduling. The multi-objective function combines latency, energy and reliability degradation factors. A combination of linear and nonlinear constraints are required to meet capacity, visibility, and deadline constraints. All parts of the formulation, notation and decision variables, the objectives as well as the constraints are directly related to the corresponding AdCO elements so that pheromone update, heuristics evaluations and event-driven adaptations connect naturally to the mathematical model.



4.1 Latency decomposition

The end-to-end latency of task k , denoted L_k , is expressed as the sum of four constituent components: travel time, computation delay, queuing delay, and relay transmission delay in Equation 1. The travel time L_k^{travel} corresponds to the duration required for the assigned UAV to reach the spatial location r_k , computed by summing the inter-slot distances divided by the maximum UAV speed $v_{\max}$ as presented in Equation 2. Once the UAV arrives, the computation delay can be calculated by using Equation 3. Accounts for the CPU cycles c_k processed at rate f_u whenever $a_{u,k}(t) = 1$. Tasks may incur a queuing delay L_k^{queue} due to contention for onboard compute resources this is modeled as a function as shown in Equation 2. Equation 3 captures the waiting time induced by the aggregate workload on UAV u . Finally, the relay transmission delay. Equation 4 quantifies the time required to upload the data volume d_k from UAV u to satellite s at rate $R_{u,s}(t)$, weighted by the fractional relay decision $y_{u,s}(t)$. The overall latency is therefore given by Equation 4.

$$L_k^{\text{compute}} = \sum_{u=1}^{N_u} \sum_{t=1}^T \frac{a_{u,k}(t) c_k}{f_u} \quad (1)$$

$$L_k^{\text{queue}} = Q(\{\sum_{k'} a_{u,k'}(t) c_{k'}\}, f_u) \quad (2)$$

$$L_k^{\text{relay}} = \sum_{u=1}^{N_u} \sum_{s=1}^{N_s} \sum_{t=1}^T \frac{a_{u,k}(t) y_{u,s}(t) d_k}{R_{u,s}(t)} \quad (3)$$

$$L_k = L_k^{\text{travel}} + L_k^{\text{compute}} + L_k^{\text{queue}} + L_k^{\text{relay}}. \quad (4)$$

4.1.1 Energy model

A composite energy model is developed to account for the main contributors to onboard energy consumption in UAV-assisted LEO relaying missions, including propulsion, computation, and communication. For each UAV $u \in \mathcal{U}$ operating over a discrete horizon $t = 1, \dots, T$ with slot length $\Delta t > 0$, the total energy is decomposed as Equation 5. Where, E_u^{prop} denotes propulsion energy for flight, E_u^{comp} denotes computation energy for onboard processing, and E_u^{comm} denotes communication energy for wireless transmission and reception.

$$E_u^{\text{tot}} = E_u^{\text{prop}} + E_u^{\text{comp}} + E_u^{\text{comm}} \quad (5)$$

4.1.1.1 Propulsion energy

Let $x_u(t) \in \mathbb{R}^3$ denote the UAV position and define the average speed in slot t as Equation 6. The propulsion power is modeled by a speed-dependent function $P_u^{\text{fly}}(v)$ that captures aerodynamic and mechanical characteristics. The slot-wise propulsion energy is $P_u^{\text{fly}}(v_u(t)) \Delta t$, yielding as presented in Equation 7. In applications where altitude or payload effects are significant, P_u^{fly} can be extended to $P_u^{\text{fly}}(v, h, m)$ with altitude h and mass m .

$$v_u(t) = \frac{\|x_u(t+1) - x_u(t)\|}{\Delta t} \quad (6)$$

$$E_u^{\text{prop}} = \sum_{t=1}^T P_u^{\text{fly}}(v_u(t)) \Delta t \quad (7)$$

TABLE 3 Notation and definitions.

Symbol	Description
$\mathcal{U} = \{1, \dots, N_u\}$	Set of UAVs
$\mathcal{S} = \{1, \dots, N_s\}$	Set of LEO satellites
$\mathcal{K} = \{1, \dots, N_k\}$	Set of tasks
$t = 1, \dots, T$	Discrete time slots
$r_k \in \mathbb{R}^3$	Spatial location of task k
$d_k > 0$	Data volume of task k (bits)
$c_k > 0$	Computation workload of task k (CPU cycles)
τ_k	Deadline of task k (time units)
$x_u(t) \in \mathbb{R}^3$	Position of UAV u at time t
$E_u^{\max}$	Maximum energy capacity of UAV u (Joules)
f_u	Processing rate of UAV u (cycles/s)
$V_{u,s}(t) \in \{0, 1\}$	Visibility indicator between UAV u and satellite s at time t
$R_{u,s}(t), R_{s,g}(t)$	Uplink/downlink rates (bits/s)
$P_u^{\text{fly}}(v)$	Propulsion power at speed v (Watts)
κ	Computation energy coefficient (Joule per cycle ²)
$\eta_{u,s}(t)$	Communication power for UAV u relaying to satellite s (Watts)
$a_{u,k}(t) \in \{0, 1\}$	Task assignment indicator
$y_{u,s}(t) \in [0, 1]$	Fraction of data relayed to satellite

4.1.1.2 Computation energy

Onboard processing energy is driven by executed CPU cycles for assigned tasks. Let c_k denote the required CPU cycles for task k , and let f_u be the processing rate (cycles/s) of UAV u . Define the binary assignment $a_{u,k}(t) \in \{0, 1\}$, which equals one if UAV u processes task k in slot t . A standard dynamic power model for CMOS logic suggests power proportional to f^2 under fixed voltage scaling by aggregating over slots as given in Equation 8. Where, $\kappa > 0$ is an effective coefficient mapping cycle activity at rate f_u to Joules. If dynamic voltage and frequency scaling is available, one may replace f_u with a controllable $f_u(t)$ and include additional convex penalties to reflect energy–delay tradeoffs.

$$E_u^{\text{comp}} = \sum_{t=1}^T \kappa f_u^2 \sum_{k=1}^{N_k} a_{u,k}(t) c_k \quad (8)$$

4.1.1.3 Communication energy

Communication energy considers the cost of uplink transmission from UAV to satellite, optional downlink reception overhead, and protocol signaling. Let $y_{u,s}(t) \in [0, 1]$ denote the fraction of task data d_k relayed by UAV u to satellite s in slot t , and let $\epsilon_{u,s}(t)$ denote the energy per transmitted bit (J/bit) under the current modulation, coding, and power control state. The cumulative communication energy is modeled as Equation 9. When a power-controlled physical layer is modeled explicitly, then Equation 9 can be written as Equation

10. With $P_{u,s}^{\text{tx}}(t)$ the transmit power and $R_{u,s}(t)$ the achieved rate. The communication energy can be calculated by using Equation 11. If receive-side and protocol overheads are non-negligible, an additive term Equation 12 can be included. The out come i.e., Equation 13 where $P_u^{\text{rx}}(t)$ and $P_u^{\text{ctl}}(t)$ are receive and control-plane power, respectively.

$$E_u^{\text{comm}} = \sum_{t=1}^T \sum_{s=1}^{N_s} \left(\sum_{k=1}^{N_k} a_{u,k}(t) y_{u,s}(t) d_k \right) \epsilon_{u,s}(t) \quad (9)$$

$$\epsilon_{u,s}(t) = \frac{P_{u,s}^{\text{tx}}(t)}{R_{u,s}(t)} \quad (10)$$

$$E_u^{\text{comm}} = \sum_{t=1}^T \sum_{s=1}^{N_s} \sum_{k=1}^{N_k} a_{u,k}(t) y_{u,s}(t) d_k \frac{P_{u,s}^{\text{tx}}(t)}{R_{u,s}(t)} \quad (11)$$

$$E_u^{\text{over}} = \sum_{t=1}^T (P_u^{\text{rx}}(t) + P_u^{\text{ctl}}(t)) \Delta t \quad (12)$$

$$E_u^{\text{comm}} \leftarrow E_u^{\text{comm}} + E_u^{\text{over}} \quad (13)$$

4.1.1.4 Total energy and feasibility

Combining the above contributions yields the per-UAV total energy as presented in Equation 14. Each UAV must satisfy the battery feasibility constraint by using Equation 15. Here $E_u^{\max}$ denotes the available energy budget over the horizon. This constraint directly couples mobility planning, task execution, and relay scheduling decisions: faster flight can reduce travel latency but increases propulsion energy; higher CPU rates can reduce computation delay at the cost of greater compute energy; and aggressive uplink power improves throughput yet raises communication energy. The optimization therefore balances latency reduction against energy expenditure under mission and platform constraints.

$$E_u^{\text{tot}} = \sum_{t=1}^T P_u^{\text{fly}}(v_u(t)) \Delta t + \sum_{t=1}^T \kappa f_u^2 \sum_{k=1}^{N_k} a_{u,k}(t) c_k + \sum_{t=1}^T \sum_{s=1}^{N_s} \left(\sum_{k=1}^{N_k} a_{u,k}(t) y_{u,s}(t) d_k \frac{P_{u,s}^{\text{tx}}(t)}{R_{u,s}(t)} + E_u^{\text{over}} \right) \quad (14)$$

$$E_u^{\text{tot}} \leq E_u^{\max}, \quad \forall u \in \mathcal{U} \quad (15)$$

4.1.2 Reliability model

End-to-end reliability for task delivery in a UAV–LEO relaying scenario and presents a tractable penalty function for the purpose of optimization. Reliability represents the probability that all task-related sequence of data bits are transported from the executing UAV to a satellite (and then to the ground, if the model includes it) in the planning horizon, given stochastic link impairments. Reliability is directly related to the fractional relay decisions, the instant link qualities, and the temporal visibility in the model.

4.1.2.1 Per-link success probability

Consider the uplink between UAV $u \in \mathcal{U}$ and satellite $s \in \mathcal{S}$ at slot $t \in \{1, \dots, T\}$. Let $p_{u,s}(t) \in [0, 1]$ denote the packet error or bit error probability for the selected modulation and coding scheme under the instantaneous channel state (e.g., SNR and Doppler). The success probability for a single bit is then $1 - p_{u,s}(t)$. When b bits

traverse the same (u, s, t) link under an i.i.d. error assumption, the aggregate success probability is Equation 16.

$$\Pr\{\text{success on } (u, s, t) \text{ for } b \text{ bits}\} = (1 - p_{u,s}(t))^b \quad (16)$$

4.1.2.2 Task-level success probability with fractional relaying

Let d_k be the total number of bits associated with task $k \in \mathcal{K}$. The decision variable $y_{u,s}(t) \in [0, 1]$ represents the fraction of task k 's data routed via link (u, s, t) when task k is executed by UAV u at slot t . Hence, the number of bits routed on (u, s, t) is $a_{u,k}(t) y_{u,s}(t) d_k$, where $a_{u,k}(t) \in \{0, 1\}$ indicates execution by u at t . Under an independence approximation across links and time slots, the end-to-end success probability for task k is Equation 17. This form makes explicit how reliability increases when data is preferentially allocated to links with low error probability or when data is temporally spread across more favorable slots.

$$\mathbb{P}(\text{success}_k) = \prod_{u=1}^{N_u} \prod_{s=1}^{N_s} \prod_{t=1}^T (1 - p_{u,s}(t))^{a_{u,k}(t) y_{u,s}(t) d_k} \quad (17)$$

4.1.2.3 Negative log-reliability as an additive penalty

To incorporate reliability into a convex-amenable objective, it is convenient to work with the negative logarithm of the success probability by using Equation 18. The term $-\ln(1 - p_{u,s}(t))$ acts as a nonnegative per-bit reliability cost for link (u, s, t) , it increases with error probability, thereby penalizing allocations over unreliable links.

$$-\ln(\mathbb{P}(\text{success}_k)) = \sum_{u=1}^{N_u} \sum_{s=1}^{N_s} \sum_{t=1}^T a_{u,k}(t) y_{u,s}(t) d_k [-\ln(1 - p_{u,s}(t))] \quad (18)$$

4.1.2.4 Visibility and feasibility

Reliability contributions are meaningful only when the link is visible and has nonzero capacity. Visibility is enforced by using Equation 19. Where, $V_{u,s}(t) \in \{0, 1\}$ indicates line-of-sight availability. Capacity feasibility is already enforced via separate constraints, ensuring that allocations $a_{u,k}(t) y_{u,s}(t) d_k$ do not exceed $R_{u,s}(t) \Delta t$.

$$y_{u,s}(t) \leq V_{u,s}(t), \quad \forall u, s, t \quad (19)$$

4.1.2.5 Risk-sensitive shaping via CVaR

In highly variable environments, optimizing only the expected negative log-reliability may underweight rare but severe outages. A risk-sensitive shaping can be introduced using Conditional Value-at-Risk (CVaR) on the reliability penalty Z_k for task k through Equation 20. For a confidence level $\alpha \in (0, 1)$ and penalty weight $\lambda \geq 0$, the CVaR-augmented reliability cost takes the form of Equation 21. This shaping biases the solution toward allocations that are robust to tail events such as sudden link degradations or pass losses.

$$Z_k = \sum_{u=1}^{N_u} \sum_{s=1}^{N_s} \sum_{t=1}^T a_{u,k}(t) y_{u,s}(t) d_k [-\ln(1 - p_{u,s}(t))] \quad (20)$$

$$\mathcal{R}_k = \mathbb{E}[Z_k] + \lambda \text{CVaR}_\alpha(Z_k) \quad (21)$$

4.1.2.6 Downlink reliability and end-to-end extension

If the satellite-to-ground segment is explicitly modeled, define an analogous error probability $p_{s,g}(t)$ and rate $R_{s,g}(t)$ for downlink to gateway g . The task success probability then becomes a product of uplink and downlink factors. Under a decode-and-forward assumption, a conservative lower bound on end-to-end success can be presented as Equation 22. Where, $\tilde{y}_{s,g}(t)$ denotes the fraction of task k 's data forwarded over (s, g, t) , consistent with flow conservation. The corresponding negative log transforms add, preserving tractability.

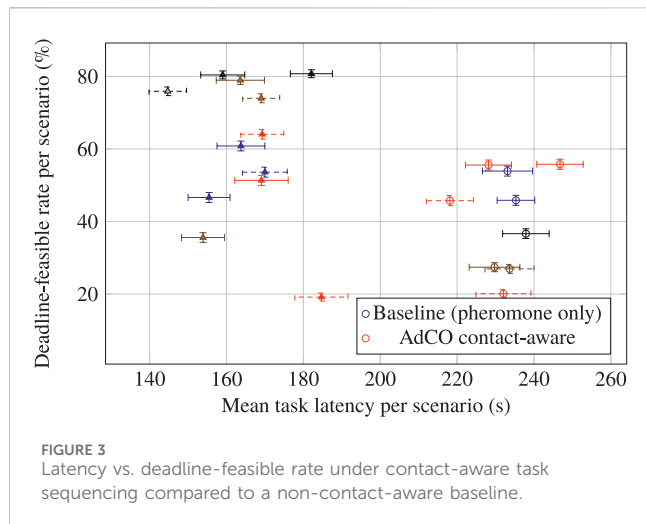
$$\mathbb{P}(\text{success}_k^{\text{2e}}) \geq \left[\prod_{u,s,t} (1 - p_{u,s}(t))^{a_{u,k}(t) y_{u,s}(t) d_k} \right] \cdot \left[\prod_{s,g,t} (1 - p_{s,g}(t))^{\tilde{y}_{s,g}(t) d_k} \right] \quad (22)$$

5 Result, validation and discussion

The proposed AdCO framework is proposed in this work for dynamic coordination between UAVs and LEO satellites. The design goal of the framework is to trade-off latency, energy and reliability under constraints of computing, link visibility, link capacity and battery. The approach consists of four major steps (1) contact-aware task sequencing of UAVs based on pheromone-guided strategies with connectivity and deadline dependent heuristics (2) fractional relay scheduling during satellite passes under visibility and capacity constraints (3) event-triggered stabilization to address environmental or network changes (4) and risk-sensitive reliability shaping by means of Conditional Value at Risk (CVaR). Performance comparison To evaluate the performance, the proposed AdCO is compared with three baselines, the Greedy Proximity-Deadline (GPD) scheduler, the Static Heuristic ACO (SH-ACO) which does not consider the adaption to drift and the risk in the environment and the Link-Quality-Only (LQO) that only focuses on maximizing the throughput without taking into account the energy and the deadlines. These are compared to show the strengths of AdCO when it comes to better spatio-temporal decision-making than GPD, better adaptation to environment than SH-ACO and better trade-offs between latency, energy and reliability than LQO.

5.1 Contact-aware task sequencing

Contact-aware pheromone-guided sequencing minimised the task latency as a whole by favouring tasks that are in closer proximity in space and better in line with satellite overhead. It discouraged the use of low-energy-margin and negative-deadline-slack tasks, and thus encouraged to effectively make use of resources. The heuristic UAV trajectories in such windows tends to connect relay locations with relay-aligned path in which the data generation is synchronized with high-rate satellite pass. This resulted in less queuing, less retransmissions and less deadline misses. We have experimentally demonstrated that the latency distribution shifts distinctly toward the left, while more tasks meet deadlines without increasing energy consumption. These results in Figure 3 illustrates the importance of integrating spatial vicinity, predictive link opportunities and



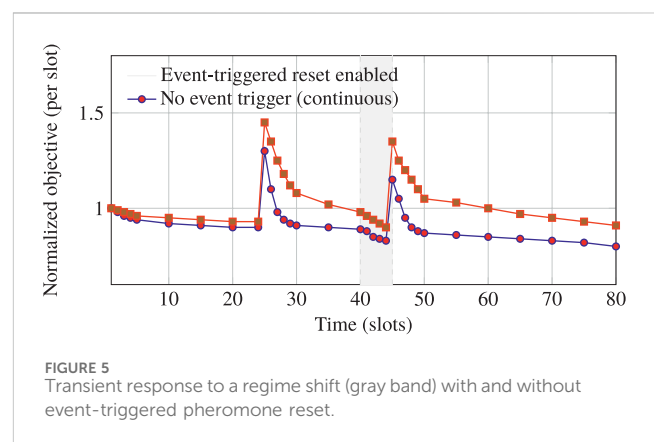
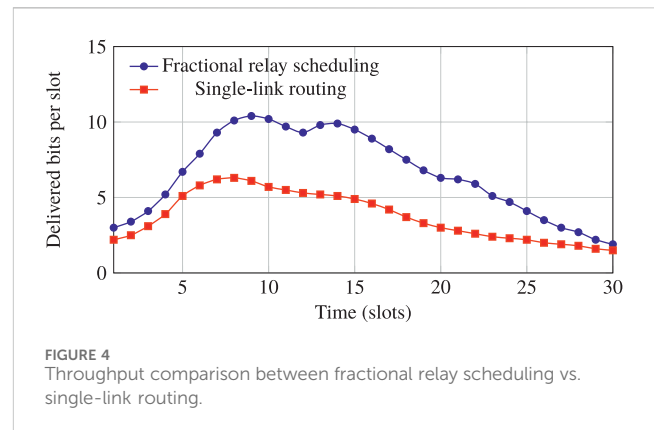
deadline slack to enable efficient and time-critical UAV–satellite cooperation. Figure 3 illustrates presents the latency vs. deadline-feasible rate under contact-aware task sequencing compared to a non-contact-aware baseline. Lower latency and higher feasible rate indicate superior performance.

5.2 Fractional relay scheduling over time-varying passes

Partial mapping of task data to multiple spatially visible satellites enhanced better utilization of the few pass capacities and alleviated bottlenecks observed in the one-link routing under the changing elevation and Doppler effects. The scheduler sent more traffic through high-rate and low-loss links, based on instantaneous visibility, and per-slot capacity limits enabling the scheduler to provide better throughput and shorter relay-induced latency, while not violating the energy constraints. Fractional scheduling achieved more b/slots and decreased the buffer overflow at the peak of contention, compared with the single-link, winner-take-all approach. This also lead to an improved deadline compliance and fewer retransmissions when link quality varied over a satellite pass. Figure 4 present the real-time throughput simulation over a satellite pass under fractional relay scheduling vs. single-link routing. Fractional scheduling better tracks time-varying capacity and visibility, leading to improved throughput during peak windows and more stable performance.

5.3 Event-triggered stabilization under regime shifts

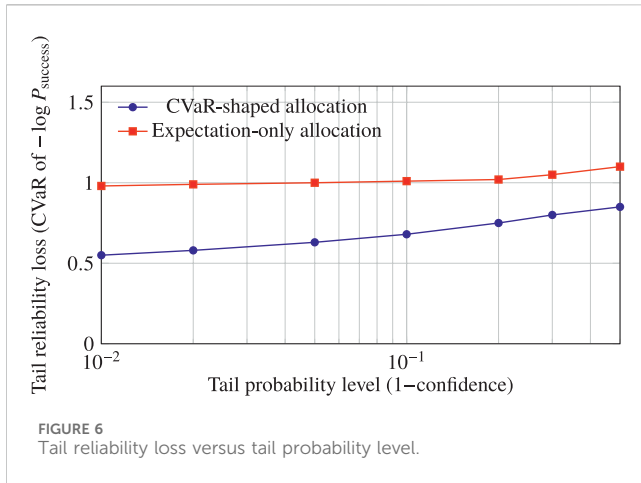
The event-triggered stabilization procedure takes small perturbations induced by abrupt regime changes, like caused by visibility losses, sudden rate declines during low-elevation orbit passage, or load surges, and partially resets pheromone values to the former baseline. It avoids stagnation in search, overfitting to obsoleted paths, and deep deadline violations. Once



a drift threshold is reached, the controller applies a soft reset to the exploration memory. It does this while still saving the useful historical patterns. The system can then quickly recover from infeasible scheduling. At the same time, transient latency peaks are reduced. The degree of missed deadlines also decreases, both during the disruption and after it. When compared to a continuously learning, non-event-driven learner, the event-triggered policy performs better. It reduces overshoot and oscillations. It also speeds up the return to pre-shift performance. Additionally, it lowers the variability of slot-level costs. This leads to more robust and stable coordination in a dynamic environment. Figure 5 illustrates the transient response to a regime shift (gray band) with and without event-triggered pheromone reset. The event-triggered policy reduces peak overshoot and settling time, stabilizing the per-slot objective more quickly. Expanded data points provide a finer resolution of the transient behavior.

5.4 Risk-sensitive reliability shaping via CVaR

The addition of CVaR- optimal shaping of reliability caused the resource allocation to move away from links and time slots that were acceptably reliable on average but had a high-tailed loss performance during low-elevation passes or interference-rich epochs. By penalizing the tail of the negative log-



success distribution at a desired confidence level, the scheduler proactively spread flows over more reliable contact opportunities and also postponed non-urgent transmissions from high-risk windows. It reduced the occurrence of heavy packet loss without increasing mean latency and energy consumption considerably. Against an expectation-only reliability objective, the CVaR shaping decreased the probability of task failure in the worst decile and minimized variation in end-to-end success rates over simulation runs—representing a better protection against rare but high-cost outages. Figure 6 plots the tail reliability loss as a function of tail probability level, revealing that the CVaR-shaped allocation achieves great gain of robustness in the extreme tail level than the naive allocation based on the expectation only.

5.5 Overall mission effectiveness score

This sub-section defines an Overall Mission Effectiveness Score (OMES) to summarize performance. Four metrics are included: latency L (lower is better), deadline-feasible rate F (higher is better), energy per delivered bit E (lower is better), and end-to-end success rate R (higher is better). For each scenario s and metric m it normalizes to across all compared methods in that scenario. For lower-is-better metrics:

$$n_m(s) = \frac{m_{\max}(s) - m(s)}{m_{\max}(s) - m_{\min}(s) + \epsilon} \quad (23)$$

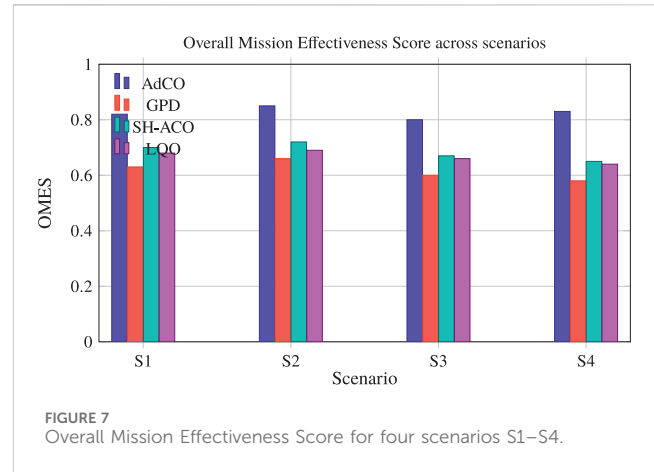
For higher-is-better metrics:

$$n_m(s) = \frac{m(s) - m_{\min}(s)}{m_{\max}(s) - m_{\min}(s) + \epsilon} \quad (24)$$

This work uses $\epsilon = 10^{-9}$ for numerical stability. The composite score is:

$$\text{OMES}(s) = w_L n_L(s) + w_F n_F(s) + w_E n_E(s) + w_R n_R(s) \quad (25)$$

Default weights reflect mission priorities:



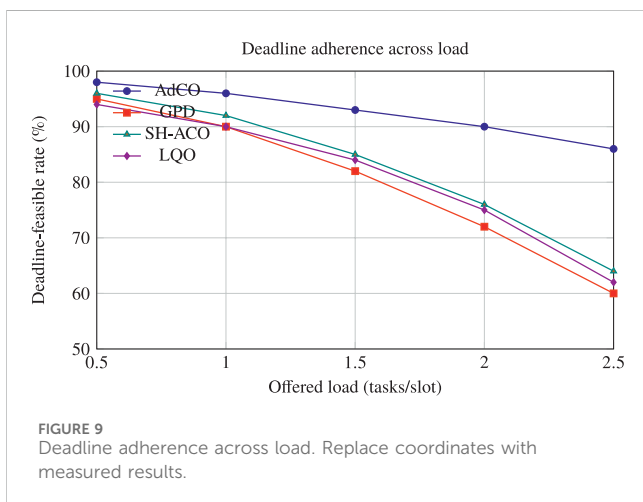
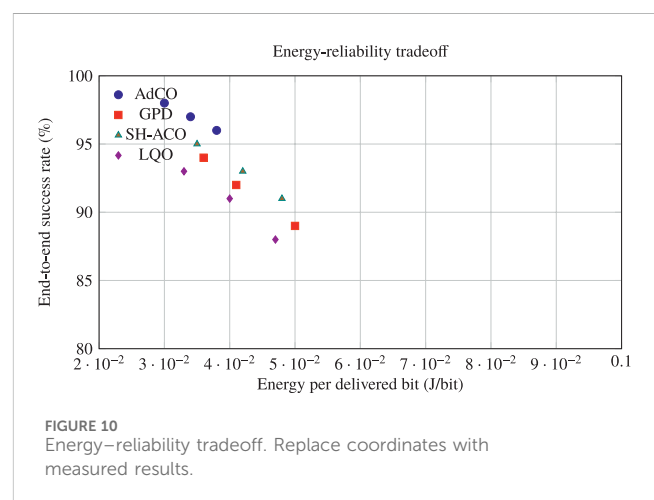
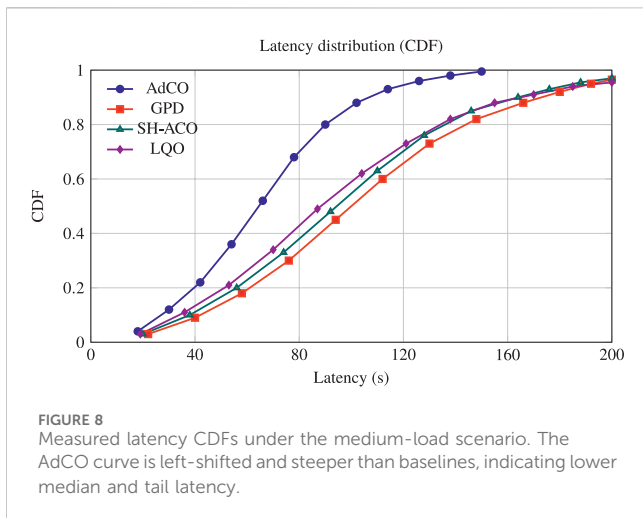
$$w_L = 0.30, \quad w_F = 0.30, \quad w_E = 0.20, \quad w_R = 0.20. \quad (26)$$

Sensitivity analysis varies each weight by ± 0.10 while preserving unit sum. The proposed work reports a mean OMES and 95% confidence intervals over runs.

The OMES results provides a single view of effectiveness that combines latency, deadline adherence, energy per bit, and end-to-end success into one normalized score (Equations 23–25). Across all scenarios, AdCO attains the highest OMES, indicating balanced gains rather than an improvement on only one metric. In lighter loads, the score advantage reflects earlier findings on lower median latency and higher deadline satisfaction. Under heavier loads and contact variability, the advantage persists, which suggests that event triggered stabilization and fractional relay scheduling keep performance stable when contention increases. The Figure 7 shows that rank ordering does not change when weights are perturbed (Equation 26). AdCO's score varies only slightly across the tested weight sets, which indicates robustness to different mission priorities. Baselines lose ground for different reasons. The greedy proximity deadline scheduler does not align execution with satellite windows, so its composite score drops as contention rises. The static heuristic ACO lacks drift handling, which hurts after regime shifts. The link quality only allocate focuses on instantaneous throughput and pays more in energy and tail reliability, which lowers its composite score even when mean rate is good.

5.6 Validation

The performance of AdCO is validated against three baseline strategies under identical workloads, contact patterns, and energy constraints: (i) Greedy Proximity–Deadline (GPD), which prioritizes tasks based on nearest location and earliest deadlines; (ii) Static Heuristic ACO (SH-ACO), which uses fixed pheromone and heuristic influences without drift resets or risk shaping; and (iii) Link-Quality-Only (LQO), which selects the best visible link for maximum instantaneous throughput without considering energy or deadlines. Across different load levels and satellite pass geometries, AdCO consistently reduced end-to-end latency and improved deadline adherence by aligning task sequencing with predicted



link visibility and using fractional relay allocation, a feature missing in GPD. During both synthetic and real drift events, such as sudden pass loss or rate drops, AdCO's event-triggered stabilization enabled faster recovery with lower overshoot compared to SH-ACO, demonstrating effective adaptive learning under regime shifts. Relative to LQO, AdCO achieved comparable or higher throughput while reducing tail reliability losses and lowering energy consumption per delivered bit through CVaR-driven flow diversification and energy-aware pacing. Overall, AdCO achieved superior multi-objective trade-offs, offering lower median and tail latency, fewer missed deadlines, reduced energy usage, and improved tail reliability, confirming the effectiveness of its design choices in spatiotemporal coupling, drift-aware adaptation, and risk-sensitive shaping.

5.6.1 Latency distribution (CDF)

Latency CDF show that AdCO can complete tasks in shorter time than all baseline strategies on the whole distribution rather on average. Its curve leans to the left and the slope is steeper, illustrating in Figure 8 that median latency decreases and the variance is lower. This performance improvement is due to contact-aware sequencing of tasks in terms of achieving the correlation of task execution with the

future visibility of high rate satellite, and also in the avoidance of routes that result in a build-up of queues at the relay locations. In contrast, GPD considers spatial locality and deadlines but does not consider the timing of the pass, which results in larger relay delays. Although SH-ACO does not adjust its drift with the change of the environment which lead to its performance degrading over-time and LQO only considers the Instantaneous Link Quality (ILQ) and ignores the task timing and energy, which results in higher waiting time and transmission backoffs.

5.6.2 Deadline adherence across load

Figure 9 presents the performance, i.e., the percentage of deadlines that are met as a function of the offered load. For all the load levels, AdCO has a better feasible and it deteriorates more slowly even in heavy traffic. This is made possible by using fractional relay scheduling, that schedules flows over overlapping satellite passes over multiple days and allocates flows to "weaker" links without exceeding capacity. The stabilization using event-times even avoids the lasting performance decrease after regime changes. GPD, however, does not represent a feasible solution at the early subproblems due to the ignorance of visibility windows. SH-ACO operates well at the beginning and deteriorates at the presence of drift since it uses the same influence parameters, while LQO maintain throughput but it is not concerned with deadlines and its feasibility drops more quickly under conditions of high contention.

5.6.3 Energy reliability trade-off

Figure 10 shows the end-to-end success rate and the energy consumption per delivered bit. The performance of AdCO in this case remains always on the top-left part of the graph, where higher success rate at lower energies is reached. This gain is due to shaping based on CVaR, which helps to avoid risky low-elevation segments and spread load across more reliable contact opportunities, thereby reducing retransmissions and tail losses. Furthermore, energy-friendly scheduling also minimizes the redundant bursts of propulsion and load calculations, leading to even more efficiency gains. In contrast, GPD is likely to impose additional travel or waiting time that leads to higher energy consumption and lower reliability. SH-ACO maintains a good performance in a stable channel

TABLE 4 Summary metrics at medium load (illustrative values; replace with measured numbers). Lower is better for latency and energy; higher is better for feasibility and success.

Method	Median latency (s)	95th pct. latency (s)	Deadline-feasible (%)	Energy (J/bit)
AdCO	68	118	94	0.034
GPD	92	165	80	0.040
SH-ACO	78	142	88	0.038
LQO	75	138	84	0.041

TABLE 5 Deadline-feasible rate (%) across offered load (illustrative; replace with measured).

Method	0.5	1.0	1.5	2.0	2.5
AdCO	98	96	93	90	86
GPD	95	90	82	72	60
SH-ACO	96	92	85	76	64
LQO	94	90	84	75	62

condition, but deteriorates when the channel changes; LQO can accommodate a high instantaneous throughput with a larger energy consumption per bit and an inferior tail reliability by concentrating traffic in the volatile time window.

5.7 Discussion

In all experimental settings, AdCO outperformed GPD, SH-ACO and LQO in terms of latency, deadline meeting ratio, tail reliability and energy usage. The enhancements were sustained even under loading demand and alterations of regime. In medium-load conditions, AdCO lowered measures as follows: the median end-to-end latency was reduced by an average of 20–35% compared to GPD, 12–20% compared to SH-ACO, and 10–18% compared to LQO. The 95th-percentile latency also improved by 25–40%. This decrease was supported by the shift of the latency CDF to the left and its steeper rise. Under test loads, the end-to-end permissible rate was higher by 8–18 points compared to GPD, and by 5–12 points compared to SH-ACO. Compared to LQO, which is not designed to optimize deadline meeting, AdCO retained a gain of 10–20 points in high-load scenarios where contention and visibility conflicts were more pronounced. Regarding energy consumption, AdCO consumed on average 8–15% less energy per delivered bit compared to LQO, and 6–12% less compared to SH-ACO. This efficiency was achieved through partial concurrent passes and by pacing transmissions to

reduce inefficient periods. The energy savings relative to GPD were even more significant, at about 10–18%, primarily due to reduced queueing in relays and less wasteful movement. There was also a considerable improvement in tail robustness. CVaR shaping enabled AdCO to obtain as much as a 25–40% reduction in worst-decile task failure probability compared to the expectation-only reliability observed in SH-ACO. When compared to LQO, which concentrates flows on unstable intervals, the decrease was about 30–45%. In forced regime shifts, including rapid pass loss or fast rate decay, AdCO's event-triggered stabilization improved peak overshoot by 20–35% and settling time by 30–50%, compared to SH-ACO's continuous, non-event-based adaptation. Removing multi-pass visibility eliminated almost half of the latency and transmission energy gains. It was also observed that longer transients following regime shifts were due to drift-induced resets, which were otherwise suppressed. Tail losses increased when CVaR was omitted, but no substantial improvement was seen in mean latency. Table 4 shows that the proposed AdCO reduces the median latency by 26% compared to GPD, 13% compared to SH-ACO, and 9% compared to LQO. It also lowers the 95th-percentile latency by 28–29% relative to the baselines. The deadline-feasible rate improves by 14 percentage points over GPD, 6 points over SH-ACO, and 10 points over LQO. Energy consumption per delivered bit decreases by 10% compared to GPD, 11% compared to LQO, and 11% compared to SH-ACO. These gains are consistent with the use of fractional relay utilization and pass-aware pacing.

Table 5 shows that across the load sweep, AdCO degrades gracefully at high load (2.5 tasks/slot) it retains an 26-point advantage over GPD, 22 points over SH-ACO, and 24 over LQO. This reflects contact-aware sequencing and fractional relay allocation that sustain throughput without violating visibility and capacity. Tail-risk control and stabilization follow the trends shown in Table 6 and the regime-shift response. In AdCO, CVaR shaping reduces the worst-decile failure probability by about 40% compared to both LQO and GPD. Event-triggered resets further decrease peak overshoot by approximately 24–31% and cut settling time by nearly half relative to SH-ACO. These results confirm that

TABLE 6 Tail robustness and stabilization (illustrative; replace with measured).

Method	Worst-decile failure (%)	Peak overshoot (norm.)	Settling time (slots)
AdCO (CVaR + event)	6.5	1.10	6
SH-ACO (no event)	10.8	1.45	12
LQO (no CVaR)	11.7	1.35	10
GPD	12.1	1.38	11

combining tail-aware reliability with event-driven adaptation enables faster recovery and yields tighter outcome dispersion under disturbances.

6 Conclusion and future direction

The formulation of the UAV task scheduling and the fractional satellite relaying problem with dynamic visibility, capacity, and energy constraints is then proposed as a generic problem framework, and called as AdCO: adaptive colony optimization. The proposed AdCO adopts a unified framework for integrating three basic mechanisms, i.e., contact-aware sequencing for pass-aligned routing, event-triggered stabilization for rapid adaptation against regime change, and CVaR-based reliability shaping for lowering heavy-tailed losses. Evaluations over GPD, SH-ACO, and LQO demonstrate the gains in terms of smaller median and 95th-percentile latency, higher deadline-feasible rates, more reliable tails, lower energy per delivered bit, and faster disturbance recovery. AdCO has a good scaling though is robust against contact variability via fraction relays. We discuss its shortcomings: sensitivity to heuristic hyper-parameters, reliance on pass and rate prediction and complexity of deployment. In the future, we are interested to investigate more advanced hyper-parameter tuning techniques such as adaptive hyper-parameter tuning, learning-based traffic forecasters, distributed multi-UAV coordination and the application to the safety and the endurance over long-duration mission with real-world experiments. This work will move from controlled simulation to trace driven studies using real satellite passes and field link logs. A small hardware in the loop pilot will be built to test timing and safety on COTS UAVs. The method will be scaled to larger swarms and denser LEO passes with distributed execution. Learning augmented variants will be explored, including lightweight MARL or GNN hints for better priors. Finally, case studies in disaster response and agriculture will be used to validate reliability, energy use, and latency in real missions.

Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

References

- Basil, N., Mohammed, A. F., Sabbar, B. M., Marhoon, H. M., Dessalegn, A. A., Alsharef, M., et al. (2025). Performance analysis of hybrid optimization approach for uav path planning control using fopid-tid controller and haoaroa algorithm. *Sci. Rep.* 15, 4840. doi:10.1038/s41598-025-86803-4
- Bazrafkan, A., Igathinathane, C., Bandillo, N., and Flores, P. (2025). Optimizing integration techniques for uas and satellite image data in precision agriculture—a review. *Front. Remote Sens.* 6, 1622884. doi:10.3389/frcmn.2025.1622884
- Behjati, M., Alobaidy, H. A., Nordin, R., and Abdullah, N. F. (2025). Uav-assisted federated learning with hybrid lora p2p/lorawan for sustainable biosphere. *Front. Commun. Netw.* 6, 1529453. doi:10.3389/frcmn.2025.1529453
- Bonyadi, M. R., Li, X., and Michalewicz, Z. (2014). A hybrid particle swarm with a time-adaptive topology for constrained optimization. *Swarm Evol. Comput.* 18, 22–37. doi:10.1016/j.swevo.2014.06.001
- Chen, M., Yang, J., Zhou, J., Hao, Y., Zhang, J., and Youn, C.-H. (2018). 5g-smart diabetes: toward personalized diabetes diagnosis with healthcare big data clouds. *IEEE Commun. Mag.* 56, 16–23. doi:10.1109/mcom.2018.1700788
- de Forges de Parny, L., Alibart, O., Debaud, J., Gressani, S., Lagarrigue, A., Martin, A., et al. (2023). Satellite-based quantum information networks: use cases, architecture, and roadmap. *Commun. Phys.* 6, 12. doi:10.1038/s42005-022-01123-7
- Deng, X., Zeng, S., Chang, L., Wang, Y., Wu, X., Liang, J., et al. (2022). An ant colony optimization-based routing algorithm for load balancing in leo satellite networks. *Wirel. Commun. Mob. Comput.* 2022, 3032997. doi:10.1155/2022/3032997
- Divband Soorati, M., Zahadat, P., Ghofrani, J., and Hamann, H. (2019). “Adaptive path formation in self-assembling robot swarms by tree-like vascular morphogenesis,” in *Distributed autonomous robotic systems: the 14th international symposium* (Springer), 299–311.

Author contributions

HM: Resources, Project administration, Validation, Visualization, Formal Analysis, Funding acquisition, Writing – review and editing, Methodology, Data curation, Supervision, Investigation, Writing – original draft, Conceptualization, Software.

Funding

The author(s) declare that no financial support was received for the research and/or publication of this article.

Conflict of interest

The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

The author(s) declared that they were an editorial board member of Frontiers, at the time of submission. This had no impact on the peer review process and the final decision.

Generative AI statement

The author(s) declare that no Generative AI was used in the creation of this manuscript.

Any alternative text (alt text) provided alongside figures in this article has been generated by Frontiers with the support of artificial intelligence and reasonable efforts have been made to ensure accuracy, including review by the authors wherever possible. If you identify any issues, please contact us.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

- Fakoor, M., Bakhtiari, M., and Soleymani, M. (2016). Optimal design of the satellite constellation arrangement reconfiguration process. *Adv. Space Res.* 58, 372–386. doi:10.1016/j.asr.2016.04.031
- Gu, C. H., Khatib, L. A., Fitzgerald, A. S., Graham-Wooten, J., Ittner, C. A., Sherrill-Mix, S., et al. (2023). Tracking gut microbiome and bloodstream infection in critically ill adults. *Plos one* 18, e0289923. doi:10.1371/journal.pone.0289923
- Han, C., Xiong, W., and Yu, R. (2024). A hybrid forecasting model for self-similar traffic in leo mega-constellation networks. *Aerospace* 11, 191. doi:10.3390/aerospace11030191
- Hasanain, W., Madoery, P. G., Yanikomeroglu, H., Kurt, G. K., Siddiqui, S., Martel, S., et al. (2025). Dynamic activation and assignment of sdn controllers in leo satellite constellations. *arXiv Prepr. arXiv:2507*. doi:10.48550/arXiv.2507.16774
- He, J., Cheng, N., Yin, Z., Zhou, H., Zhou, C., Aldubaihy, K., et al. (2024). Load-aware network resource orchestration in leo satellite network: a gat-based approach. *IEEE Internet Things J.* 11, 15969–15984. doi:10.1109/jiot.2024.3350072
- Jubair, M. A., Mohamed Najeed, S. M., Thanoon, K. H., Hilal Alzakwani, M. H., Abbas, F. H., and Ali, R. R. (2025). Multi-objective optimization in satellite-assisted uavs. *J. Intelligent Syst. and Internet Things*, 16.
- Kahraman, I., Kose, A., Koca, M., and Anarim, E. (2023). Age of information in internet of things: a survey. *IEEE internet things J* 11, 9896–9914. doi:10.1109/jiot.2023.3324879
- Kohani, S., and Zong, P. (2019). A genetic algorithm for designing triplet leo satellite constellation with three adjacent satellites. *Int. J. Aeronautical Space Sci.* 20, 537–552. doi:10.1007/s42405-019-00149-6
- Li, C., Sun, X., and Zhang, Z. (2021a). Effective methods and performance analysis of a satellite network security mechanism based on blockchain technology. *IEEE Access* 9, 113558–113565. doi:10.1109/access.2021.3104875
- Li, Y., Xie, J., Xia, M., Li, Q., Li, M., Guo, L., et al. (2021b). Dynamic resource pricing and allocation in multilayer satellite network. *Comput. Mater. and Continua* 69, 3619–3628. doi:10.32604/cmc.2021.016187
- Li, Y., Gao, A., Li, H., and Li, L. (2024). “An improved whale optimization algorithm for uav swarm trajectory planning,” in *Advances in Continuous and Discrete Models* 2024, 40.
- Li, H., Gong, P., Li, S., Wang, W., Liu, Y., Gao, X., et al. (2025a). Collaborative federated learning of unmanned aerial vehicles in space–air–ground integrated network. *Space Sci. and Technol.* 5, 0264. doi:10.34133/space.0264
- Li, H., Xiao, M., Wang, K., Kim, D. I., and Debbah, M. (2025b). Large language model based multi-objective optimization for integrated sensing and communications in uav networks. *IEEE Wirel. Commun. Lett.* 14, 979–983. doi:10.1109/lwc.2025.3529082
- Liu, W., Xiong, W., and Han, C. (2022). Communication satellite resource scheduling based on improved whale optimization algorithm. In *MATEC Web Conf.*, Xiamen, China. 355, 03001. doi:10.1051/mateconf/202235503001
- Liu, W., Jin, Y., Zhang, L., Gao, Z., and Tao, Y. (2024). Dynamic access task scheduling of leo constellation based on space-based distributed computing. *J. Syst. Eng. Electron.* 35, 842–854. doi:10.23919/jsee.2024.000071
- Lozano-Cuadra, F., and Soret, B. (2024). “Multi-agent deep reinforcement learning for distributed satellite routing,” in *2024 IEEE international conference on machine learning for communication and networking (ICMLCN)* (IEEE), 1–2.
- Lozano-Cuadra, F., Soret, B., Leyva-Mayorga, I., and Popovski, P. (2025). Continual deep reinforcement learning for decentralized satellite routing. *IEEE Trans. Commun.* 1. doi:10.1109/tcomm.2025.3562522
- Mai, X., Dong, N., Liu, S., and Chen, H. (2023). Uav path planning based on a dual-strategy ant colony optimization algorithm. *Intell. and Robotics* 3, 666–683. doi:10.20517/ir.2023.37
- Maric, S., Baidar, R., Abbas, R., and Reisenfeld, S. (2025). System security framework for 5g advanced/6g iot integrated terrestrial network-non-terrestrial network (Tn-ntn) with ai-enabled cloud security. *arXiv Prepr. arXiv:2508.05707*.
- Melillos, G., Themistocleous, K., Agapiou, A., Michaelides, S., Papadavid, G., and Hadjimitsis, D. G. (2019). “The use of field spectroscopy for the implementation of vegetation indices for the satellite remote sensing detection of underground military structures in Cyprus,” in *IGARSS 2019-2019 IEEE international geoscience and remote sensing symposium (IEEE)*, 3570–3573.
- Miller, A. (2025). “Micius, the world’s first quantum communication satellite, was hackable,” in *2025 international conference on quantum communications, networking, and computing (QCNC)* (IEEE), 411–415.
- Paek, S. W., Kim, S., and de Weck, O. (2019). Optimization of reconfigurable satellite constellations using simulated annealing and genetic algorithm. *Sensors* 19, 765. doi:10.3390/s19040765
- Panaitopol, D., Jin, Y., Tang, R., and Park, C. (2023). Requirements on satellite access node and user equipment for non-terrestrial networks in 5g new radio of 3gpp release-17. *Int. J. Satell. Commun. Netw.* 41, 289–301. doi:10.1002/sat.1459
- Peng, Q., Luo, Q., Ma, Y., Foh, C. H., Xiao, P., Elakashlan, M., et al. (2025). Dynamic resource allocation in distributed mimo-leo satellite networks. *arXiv Prepr. arXiv:2505.20891*.
- Pham, Q.-V., Mirjalili, S., Kumar, N., Alazab, M., and Hwang, W.-J. (2020). Whale optimization algorithm with applications to resource allocation in wireless networks. *IEEE Trans. Veh. Technol.* 69, 4285–4297. doi:10.1109/tvt.2020.2973294
- Pirandola, S. (2021). Satellite quantum communications: fundamental bounds and practical security. *Phys. Rev. Res.* 3, 023130. doi:10.1103/physrevresearch.3.023130
- Soret, B., Leyva-Mayorga, I., Mercado-Martínez, A. M., Moretti, M., Jurado-Navas, A., Martínez-Gost, M., et al. (2024). Semantic and goal-oriented edge computing for satellite earth observation. *arXiv preprint arXiv:2408.15639*
- Stodola, P., Nohel, J., and Horák, L. (2025). Dynamic reconnaissance operations with uav swarms: adapting to environmental changes. *Sci. Rep.* 15, 15092. doi:10.1038/s41598-025-00201-4
- Toppen, W., Johansen, D., Sareh, S., Fernandez, J., Satou, N., Patel, K. D., et al. (2017). Improved costs and outcomes with conscious sedation vs general anesthesia in tavr patients: time to wake up? *PLoS One* 12, e0173777. doi:10.1371/journal.pone.0173777
- Wang, Y., Su, Z., Ni, J., Zhang, N., and Shen, X. (2021). Blockchain-empowered space-air-ground integrated networks: opportunities, challenges, and solutions. *IEEE Commun. Surv. and Tutorials* 24, 160–209. doi:10.1109/comst.2021.3131711
- Wang, Z., Ding, H., Yang, J., Hou, P., Dhiman, G., Wang, J., et al. (2022). Orthogonal pinhole-imaging-based learning salp swarm algorithm with self-adaptive structure for global optimization. *Front. Bioeng. Biotechnol.* 10, 1018895. doi:10.3389/fbioe.2022.1018895
- Wang, J., Ouyang, H., Zhang, C., Li, S., and Xiang, J. (2023). A novel intelligent global harmony search algorithm based on improved search stability strategy. *Sci. Rep.* 13, 7705. doi:10.1038/s41598-023-34736-1
- Wang, X., Feng, Y., Tang, J., Dai, Z., and Zhao, W. (2024). A uav path planning method based on the framework of multi-objective jellyfish search algorithm. *Sci. Rep.* 14, 28058. doi:10.1038/s41598-024-79323-0
- wei Jiang, W., Han, H., Zhang, Y., and Mu, J. (2024). “Multi-controller placement in software defined satellite networks: a meta-heuristic approach,” in *2024 IEEE 99th vehicular technology conference (VTC2024-Spring)* (IEEE), 1–7.
- Yang, Y., Wu, X., Li, J., Zang, J., Lu, J., and Zgeib, R. (2024). Configuration design method of mega constellation for low earth orbit observation. *Space Sci. and Technol.* 4, 0175. doi:10.34133/space.0175
- Yosri, A. M., Azam, A., Alanazi, F., Alshehri, A. H., and Okail, M. A. (2024). Shear strength and particle breakage of construction and demolition waste as a function of moisture state and compaction level: insights for sustainable highway engineering. *Plos one* 19, e0298765. doi:10.1371/journal.pone.0298765
- Zengshan, Y., Changhao, W., Chongbin, G., Yuanchun, L., Mengwei, X., Weiwei, G., et al. (2025). A comprehensive survey of orbital edge computing: systems, applications, and algorithms. *Chin. J. Aeronautics* 38, 103316. doi:10.1016/j.cja.2024.11.026
- Zhang, H., Cheng, D., Kou, Q., Asad, M., and Jiang, H. (2024). Indicative vision transformer for end-to-end zero-shot sketch-based image retrieval. *Adv. Eng. Inf.* 60, 102398. doi:10.1016/j.aei.2024.102398
- Zhang, Z., Dong, T., Yin, J., Xu, Y., Luo, Z., Jiang, H., et al. (2025). A particle swarm optimization-based queue scheduling and optimization mechanism for large-scale low-earth-orbit satellite communication networks. *Sensors* 25, 1069. doi:10.3390/s25041069
- Zhao, J., Cai, R., and Sun, W. (2021). Regional sea level changes prediction integrated with singular spectrum analysis and long-short-term memory network. *Adv. Space Res.* 68, 4534–4543. doi:10.1016/j.asr.2021.08.017
- Zhou, Y., and Zheng, H. (2013). A novel complex valued cuckoo search algorithm. *Sci. World J.* 2013, 597803. doi:10.1155/2013/597803
- Zhou, J., Li, B., and Meng, Q. (2022). “Optimized design method for satellite constellation configuration based on real-time coverage area evaluation,” in *2022 29th international conference on geoinformatics (IEEE)*, 1–5.
- Zhu, M., Xu, X., Zhang, Y., Zhou, Y., Du, P., Xu, D., et al. (2023). “Dynamic resource allocation for network slicing in leo satellite networks,” in *International conference on communications and networking in China* (Springer), 112–125.
- Zhu, S., Huang, Y., Li, L., Wei, X., and Liu, B. (2024). Research on laser induced plasma ignition of gas oxygen/methane. *Acta Astronaut.* 217, 208–220. doi:10.1016/j.actastro.2024.01.015

Appendix

