vehicles

MDPI

Article

Adaptive Mode-Aware Swarm Hybrid Optimization for Heterogeneous UAV-Assisted IoV-MEC Networks

Yeonwoo Lee

Department of Artificial Intelligence Engineering, Mokpo National University, Muan-Gun Chonnam 58554, Republic of Korea; ylee@mokpo.ac.kr; Tel.: +82-61-450-2746

Abstract

UAV-mounted Mobile Edge Computing (MEC) can extend vehicular computing where terrestrial infrastructure is sparse, but dense traffic and limited UAV batteries make resource management difficult. We consider a swarm in which a base-station-connected Head-UAV coexists with a flying ad hoc network of Member-UAVs and propose Adaptive Mode-Aware Swarm Hybrid (A-MASH), which jointly decides vehicle association with admission control, MEC CPU allocation, uplink power and UAV-to-UAV relaying. A battery- and load-aware ant colony search over the discrete decisions is coupled with a Grey Wolf Optimizer over the continuous resources; the Head-UAV access mode follows load, battery and interference-cancellation quality, and a Kalman-filter channel predictor supports decoding. In slot-level simulations with 10–150 vehicles and equal evaluation budgets, A-MASH delivers 65–102% more tasks on time than standalone metaheuristics at 100 vehicles and matches an alternating-optimization baseline up to 100 vehicles, but simple model-based heuristics are better once the load exceeds the computing capacity of the swarm. The predictor adds up to 5.5 percentage points of delivery ratio, NOMA helps only with near-perfect interference cancellation, and the battery-aware mechanisms postpone the first UAV withdrawal by one scheduling epoch.

Keywords: Unmanned Aerial Vehicle (UAV); Internet of Vehicles (IoV); Mobile Edge Computing (MEC); NOMA/OFDMA; GWO–ACO Hybrid Optimization

1. Introduction

The rapid evolution of autonomous driving and the Internet of Vehicles (IoV) has led to unprecedented growth in computationally intensive and latency-sensitive on-board services such as cooperative perception, real-time high-definition (HD) map fusion, and collision-avoidance decisions. However, the limited on-board computational resources of individual vehicles are insufficient to handle such massive data in real time. These applications require substantial computation and low end-to-end latency, which may exceed the processing capability of conventional vehicle on-board units [1,2].

To overcome this challenge, Mobile Edge Computing (MEC) has emerged as a key solution by pushing computational capabilities to the network edge [2]. MEC servers physically deployed close to the radio access network and end users therefore reduce task-offloading latency and alleviate the limitations of vehicle-side computing [1,2]. Conventional MEC infrastructures are primarily deployed at terrestrial network nodes, such as base stations or access points. However, the spatial coverage of terrestrial Base Stations (BSs) can be limited or uneven in rural roads, mountainous regions, and disaster zones, where communication and computing demands may also vary dynamically [1,3]. In particular, mounting MEC

Check for updates

Academic Editor: Mohammed Chadli

Received: 30 September 2026
Revised: 6 November 2026
Accepted: 8 November 2026
Published: 10 November 2026

Copyright: © 2026 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

servers on Unmanned Aerial Vehicles (UAVs) has attracted considerable attention, since UAVs overcome the coverage limitations of terrestrial BSs and can quickly provide flexible, re-deployable computing and communication support where terrestrial infrastructure is limited [1–3]. Moreover, air-to-ground (A2G) links between vehicles and UAVs generally exhibit a high probability of Line-of-Sight (LoS) propagation [4].

Most existing studies of UAV-assisted MEC systems consider a single-mode architecture. They consider either an infrastructure-connected UAV architecture with access to terrestrial networks [1–3] or a decentralized flying ad hoc network (FANET) architecture in which UAVs cooperatively self-organize [5]. However, relatively limited attention has been given to architectures that explicitly integrate these two operational roles within a single heterogeneous UAV-MEC swarm [1–3,5]. A single-mode deployment cannot serve dense central corridor traffic and wide-area peripheral traffic equally well. This motivates a heterogeneous design in which a high-capacity, BS-connected Head-UAV serves dense central traffic, while a distributed FANET-based Member-UAV swarm supports wider-area peripheral traffic.

Non-Orthogonal Multiple Access (NOMA) can improve spectral efficiency and connectivity in dense uplinks [6,7], but its performance is sensitive to imperfect Successive Interference Cancellation (SIC). In high-speed NOMA networks, the relative geometry between vehicles and the UAV changes rapidly enough to cause channel gain crossover within short time intervals, inverting the SIC decoding order assumed when power coefficients were assigned [8,9]. Therefore, existing mobility-aware NOMA schemes have considered user dissociation, reassociation, and re-clustering according to changing channel conditions [8,9]. These observations motivate the prediction-assisted SIC support introduced later in A-MASH.

Bio-inspired metaheuristics such as the Grey Wolf Optimizer (GWO) [10], and Ant Colony Optimization (ACO) [11] provide flexible search mechanisms for complex nonlinear optimization problems. However, standalone metaheuristics may exhibit limitations in balancing exploration and exploitation, motivating hybrid GWO-ACO strategies that combine complementary search behaviours [12]. Mixed discrete–continuous problems of this kind are commonly solved by alternating optimization, which fixes one block, optimizes the other and stops at the first fixed point. Existing hybrid GWO–ACO schemes instead run both searches on one continuous encoding [12]. Neither approach lets the continuous solution steer how new discrete candidates are built. UAV energy consumption and limited battery capacity are also critical design constraints in aerial networks [13,14]. However, minimizing total energy consumption itself does not necessarily prevent uneven battery depletion across a cooperative UAV swarm. Moreover, in dense traffic the offered computing load can exceed the capacity of the whole swarm, so that no allocation meets every deadline; a practical scheme must then decide which tasks to admit.

Motivated by the above challenges, this paper considers a structurally heterogeneous dual-mode UAV-assisted IoV-MEC network in which a BS-connected Head-UAV coexists with a FANET-based swarm of Member-UAVs. We propose A-MASH, an Adaptive Mode-Aware Swarm Hybrid optimization framework that jointly optimizes vehicle-to-UAV association with admission control, MEC CPU allocation, uplink power allocation and optional UAV-to-UAV (U2U) relay decisions. The Head-UAV access state is set by a rule that accounts for load, battery and SIC quality, whereas the Member-UAVs use orthogonal frequency-division multiple access (OFDMA). A-MASH integrates a Kalman-Filter-based Prediction-assisted SIC ordering and Rate selection (KF-PSR) receiver, a bidirectional GWO–ACO information exchange and Overflow/Rescue workload redistribution. The main contributions are summarized as follows:

- Heterogeneous dual-mode UAV-MEC architecture: a high-capacity BS-connected Head-UAV (Mode A, $F_0^{\max} = 15$ GHz) with SIC-aware NOMA/OFDMA coexists with OFDMA Member-UAVs forming a FANET (Mode B, $F_k^{\max} = 10$ GHz). Overflow and Rescue redistribute pending workloads between the two domains according to MEC utilization and the battery level predicted for the end of each epoch.
- Hybrid GWO–ACO with Bi-IE: a load- and battery-aware ACO searches the association, admission and relay decisions, GWO refines the CPU and power allocations, and the Bidirectional Information Exchange (Bi-IE) couples the two searches through residual-capacity guidance and warm-started re-allocation. An ablation shows that the iterative coupling is the main source of the gain.
- Formulation and evaluation methodology: the joint problem is formulated as a soft-deadline MINLP with admission control, and all methods are compared under an equal budget of objective evaluations in a slot-level simulator with vehicle mobility, correlated shadowing, CSI delay and an implemented Kalman-filter predictor, with significance tests, latency distributions, a mission-lifetime study and a run-time analysis.

Compared with the alternatives considered in this paper, A-MASH offers four practical advantages. First, it treats the discrete decisions (association, admission and relaying) and the continuous ones (CPU and power) with search operators suited to each, instead of forcing all of them into one continuous encoding. Second, its load-aware construction and the Bi-IE coupling find good solutions early and keep improving with more evaluations: under the same budget it delivers 65–102% more tasks on time than standalone metaheuristics at 100 vehicles. Third, it needs no training and re-optimizes every 30 s epoch in about 2 s, so it adapts directly to changes in traffic, channel and battery state. Fourth, admission control, battery-aware assignment and channel prediction act in one framework, which keeps the 95th-percentile latency close to the deadline, avoids emergency handovers between UAVs and reduces outages under mobility. Its limitation under traffic loads beyond the computing capacity of the swarm is analyzed in Section 5.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 presents the heterogeneous dual-mode network architecture and the formulation of the joint mixed-integer nonlinear programming (MINLP) problem. Section 4 introduces the proposed A-MASH framework, its four-phase optimization procedure and its complexity. Section 5 presents the simulation setup, the comparison with the baselines, and the lifetime, ablation and sensitivity analyses. Section 6 concludes the paper.

2. Related Work

UAV-assisted MEC has been extensively investigated as a means of bringing computation closer to users in dynamic and infrastructure-limited environments [1,2]. Mobility-aware task-offloading frameworks for UAV-MEC-assisted IoV have been proposed in [1] to account for vehicle mobility and reduce end-to-end latency. Barick and Singhal [15] proposed a collaborative UAV-assisted MEC architecture in which multiple UAVs cooperate with terrestrial edge servers for task offloading in urban Internet of Things (IoT) environments. Comprehensive surveys [2] have identified UAV trajectory, user association, offloading decisions, and computation-resource allocation as key design dimensions in UAV-assisted MEC systems. Wang et al. [16] further considered joint task offloading and migration in dynamic UAV-MEC networks and formulated adaptive task scheduling using an imitation-learning-based framework. However, these studies do not explicitly model the coexistence of a persistent BS-connected Head-UAV and a FANET-based Member-UAV swarm within a single heterogeneous MEC architecture. This motivates the heterogeneous dual-mode architecture considered in this study.

FANETs enable direct UAV-to-UAV (U2U) communication and distributed cooperation without continuous dependence on terrestrial infrastructure [5]. Bekmezci et al. [5] introduced the fundamental FANET architecture and major communication challenges, while subsequent surveys [17,18] reviewed routing protocols, mobility models, and protocol-stack design, clustering, and related networking issues. Recent studies have also identified limited UAV battery capacity and energy replenishment as important constraints on long-duration FANET operation [14]. Although these studies demonstrate the advantages of distributed FANET operation, FANET communication and UAV-MEC offloading are often investigated separately. FANET studies have largely focused on routing, clustering, and swarm connectivity [5,17,18], whereas UAV-MEC studies have primarily addressed task offloading, computation, and resource allocation [1,2,15,16]. Therefore, relatively limited attention has been given to architectures in which a high-capacity BS-connected UAV cooperates with autonomous FANET-based Member-UAVs while jointly considering MEC load and battery conditions. In the proposed A-MASH, the Member-UAV FANET complements the infrastructure-connected Head-UAV through distributed OFDMA, MEC processing, and optional U2U forwarding.

Joint resource allocation in UAV-MEC systems generally involves non-convex and mixed discrete–continuous decision spaces [2,3,7], motivating the use of bio-inspired metaheuristics. GWO provides a population-based search mechanism that balances exploration and exploitation [10], while ACO-based methods have been applied to heterogeneous multi-UAV coverage-path planning [19] and FANET inter-cluster routing [20]. Alternating optimization, which fixes one block of variables and optimizes the other until a fixed point is reached, is the classical model-based approach to such problems [7]. Hybrid GWO-ACO approaches have also been investigated to improve the balance between exploration and exploitation [12]. More recently, adaptive GWO variants such as HSB-GWO have been developed for cooperative multi-UAV path planning [21]. Multi-objective bio-inspired optimization has also been applied to energy-efficient and load-balanced FANET clustering [22]. These methods address individual aspects of UAV optimization, but they do not jointly couple continuous MEC/NOMA resource allocation with discrete vehicle association and U2U relay decisions while explicitly considering swarm battery balance. The proposed A-MASH framework addresses this coupling through GWO-ACO Bidirectional Information Exchange (Bi-IE).

NOMA has been actively investigated in UAV and vehicular MEC networks because power-domain multiplexing enables multiple users to share the same time–frequency resources [6,7,23]. However, NOMA performance is sensitive to SIC-order consistency. In high-mobility IoV environments, rapid variations in vehicle-UAV channel conditions may change the relative channel-gain ordering of NOMA users and degrade SIC performance. Mobility-aware NOMA user re-clustering and reassociation mechanisms have been investigated in [8,9] to address mobility-induced changes in user grouping and channel conditions. These studies demonstrate the importance of mobility-aware adaptation when user grouping and channel conditions change rapidly. This motivates the use of prediction-assisted support for NOMA operation in highly dynamic vehicular environments. Position and mobility information have also been exploited to support communication decisions in dynamic vehicular environments. Qiu et al. [24] investigated position-information-based NOMA for mobile downlink and uplink transmission, while Jaiswal [25] employed Kalman-filter-based prediction as a mobility-support mechanism in vehicular ad hoc networking.

Learning-based and game-theoretic schemes form a further class. Li et al. [26] combine deep reinforcement learning with game theory for computation offloading in dynamic edge-computing markets, and Li et al. [27] couple multi-agent proximal policy optimization (MAPPO) with an exact potential game for task offloading and resource allocation in

high-altitude platform station (HAPS)-assisted low-Earth-orbit (LEO) satellite networks. These methods learn an offloading policy through training or reach an equilibrium among self-interested agents. A-MASH instead performs a per-epoch model-based search that needs no training and re-optimizes directly when the channel, load or battery state changes. A fair comparison with such methods would need a training budget and a protocol matched to the per-decision evaluation budget used in this paper; we therefore leave it to future work (Section 6).

Motivated by these observations, A-MASH incorporates KF-PSR, a Kalman-filter channel predictor that combines the predicted vehicle position with a filtered shadowing estimate. At the Head-UAV it sets the SIC decoding order and the rate targets of each NOMA cluster from gains predicted for the decoding instant, and at all UAV receivers it is used for rate selection. It does not modify the vehicle-to-UAV association. As summarized in Table 1, the reviewed studies do not combine this type of prediction-assisted SIC support with the heterogeneous Head-UAV/FANET Member-UAV MEC architecture considered in this work.

Table 1. Comparison of the proposed A-MASH with representative recent works.

Reference	Mode/Arch.	Access Scheme	Optimization	Predictive	Battery Balance
Wang et al. [16]	Multi-UAV MEC/ Single infra.	OFDMA	Imitation learning	No	No
Peng et al. [3]	Three-tier UAV-MEC	OFDMA	Stochastic/convex optimization	No	No
Liu et al. [23]	Multi-UAV collaborative MEC/ FANET	NOMA	TD3-based DRL	No	Partial
Wu et al. [7]	UAV-enabled MEC/ Single infra.	NOMA/ FDMA/TDMA	Alternating optimization	No	No
Zheng et al. [21]	Cooperative multi-UAV/ FANET	—	Adaptive GWO	No	No
Li et al. [26,27]	Edge market/ HAPS-LEO	—	DRL, MAPPO + game theory	No	No
Proposed A-MASH	Heterogeneous dual-mode	Head: SIC-aware NOMA/OFDMA; Members: OFDMA	Hybrid GWO–ACO with Bi-IE and admission	Yes (KF-PSR, slot-level evaluation)	Yes (predictive BLI, variance)

3. System Model and Problem Formulation

3.1. Dual-Mode Mode A/Mode B Coexistence Network Architecture

We consider a UAV-assisted Internet of Vehicles (IoV) scenario in which V vehicles moving on an urban road grid offload computationally intensive and latency-sensitive tasks to a swarm of K UAVs equipped with MEC servers. The UAV swarm operates within a 2 km × 2 km service area at a uniform altitude of $h = 100$ m. The UAV set is denoted by $\mathcal{U} = \{u_k \mid k \in \mathcal{K}\}$, where $\mathcal{K} = \{0, \ldots, K-1\}$ and the vehicle set is denoted by $\mathcal{V} = \{v_i \mid i \in \mathcal{I}\}$, where $\mathcal{I} = \{1, \ldots, V\}$.

Conventional UAV-assisted MEC systems typically depend either on a centralized infrastructure-based UAV architecture, which may create a traffic and computation bottleneck, or on a decentralized FANET architecture, which lacks direct access to terrestrial infrastructure. However, neither architecture fully represents urban UAV-IoV deployments,

in which an infrastructure-connected aerial gateway may coexist with a decentralized UAV swarm. To overcome this operational asymmetry, the proposed A-MASH architecture partitions the UAV swarm into two structurally different domains: a single BS-connected Head-UAV operating in the Mode-A infrastructure domain and multiple Member-UAVs operating in the Mode-B FANET domain. These two groups differ in MEC capacities, connectivity characteristics, and default access scheme, but cooperate through workload redistribution when congestion or battery depletion occurs.

$$\mathcal{U} = \{u_0\}_{Head-UAV\ :\ Mode\ A} \cup \{u_k \mid k \in \mathcal{K} \setminus \{0\}\}_{Member-UAVs:\ Mode\ B}. \tag{1}$$

Importantly, the structural domain of a UAV is distinguished from its instantaneous access state. The Head-UAV is NOMA-capable and may switch between NOMA and OFDMA according to traffic density, battery condition and SIC quality, whereas the Member-UAVs always use OFDMA within the FANET domain.

Head-UAV (u_0): Infrastructure-Connected Node (Mode A). The Head-UAV u_0 is located near the centre of the service area and maintains a continuous backhaul link connected to a ground Base Station (BS), and it is equipped with a high-capacity MEC server with maximum CPU frequency, $F_0^{max} = 15$ GHz to handle the dense traffic concentrated in the central service region. The backhaul between the Head-UAV and the BS carries only control information and task results, because every admitted task is computed on board the UAVs; we therefore assume that it is not a bottleneck and do not model it. The Head-UAV's uplink access state $m_0 \in \{NOMA,\ OFDMA\}$ is updated at the end of each scheduling epoch (Phase 4) as

$$m_0 = \begin{cases} NOMA, & |\ \mathcal{I}_0\ | \geq \rho_{NOMA}\ \wedge\ BLI_0 \geq 0.30 \wedge \hat{\varepsilon}_{SIC} \leq \varepsilon_{th} \\ OFDMA, & \text{otherwise,} \end{cases} \tag{2}$$

where $\mathcal{I}_0 = \{i \in \mathcal{I} | a_i = 0\}$ is the set of vehicles associated with the Head-UAV, $\rho_{NOMA} = 20$ is the traffic threshold for NOMA activation, BLI_0 is the Head-UAV battery lifetime index (Section 3.4), and $\hat{\varepsilon}_{SIC}$ is the residual SIC factor estimated for the Head-UAV receiver. The SIC condition reflects the order-dependent NOMA model of Section 3.2: NOMA improves on OFDMA only when SIC is (near-)perfect, and $\varepsilon_{th} = 0.0025$ lies in the middle of the break-even interval ($0 < \hat{\varepsilon}_{SIC} < 0.005$) measured in Section 5.5. Vehicle speed (60–120 km/h) does not gate NOMA activation; its effect on SIC is handled by the predictor described next.

When $m_0 = NOMA$, the Head-UAV groups its users into NOMA clusters (Section 3.2) and decodes each cluster by Successive Interference Cancellation (SIC). Under high mobility, the channel gains of the users in a cluster change between the moment the receiver measures them and the moment it decodes, so a decoding order and rate targets chosen from stale measurements can be wrong. The Head-UAV therefore uses the KF-PSR mechanism of Section 4.1.1, which predicts each user's channel gain a few slots ahead and sets the SIC order and the rate targets from the predictions. KF-PSR operates within the current Head-UAV user set; it does not change the association a_i or the membership of $\mathcal{I}_0$. The same predictor is used at every UAV receiver for rate selection, including in OFDMA mode. Table 2 summarizes the policy.

Table 2. Head-UAV NOMA activation and KF-PSR operation under high-mobility conditions (60–120 km/h).

Aspect	Description
Head-UAV structural role	Mode-A infrastructure-connected UAV; NOMA-capable
NOMA activation condition	m_0 = NOMA when $\|\ \mathcal{I}_0\ | \geq \rho_{NOMA}$, $BLI_0 \geq 0.30$ and $\hat{\varepsilon}_{SIC} \leq \varepsilon_{th}$; otherwise $m_0 = OFDMA$
Vehicle speed	60–120 km/h; speed does not gate NOMA activation

Table 2. *Cont.*

Aspect	Description
NOMA clustering	$G = 2$ users per cluster on orthogonal sub-bands; at most $G - 1$ cancellations per cluster
KF-PSR operation	Predicts each gain L slots ahead (Kalman filter on position + first-order autoregressive filter on shadowing); sets SIC order and rate targets; used for rate selection at all UAVs

Member-UAVs ($u_1, \ldots, u_{K-1}$): Autonomous Swarm Mode (Mode B). The Member-UAVs u_k, $k \in \mathcal{K} \setminus \{0\}$ are distributed around the peripheral region of the service area and form a self-organizing flying ad hoc network (FANET). Each Member-UAV is equipped with a standard MEC server ($F_k^{\max} = 10$ GHz) and communicates with neighbour UAVs through UAV-to-UAV (U2U) links for cooperative relaying and load distribution. Because peripheral traffic density is generally lower than in the central region, the Member-UAVs use OFDMA exclusively, avoiding SIC overhead while providing sufficient per-user bandwidth via orthogonal subcarriers:

$$m_k = OFDMA,\ k \in \mathcal{K} \setminus \{0\}. \tag{3}$$

Therefore, OFDMA users transmit at full power p_{max}, and the power coefficient α_i^p is used only by Head-UAV NOMA users. For a vehicle associated with Member-UAV u_k, the Phase-2 ACO stage jointly determines the serving-UAV association a_i and the optional relay decision r_i. When $r_i = -1$, the task is processed without U2U forwarding; otherwise, a fraction ξ of the task (data and computation) is forwarded from the serving Member-UAV to another feasible UAV through a U2U link. This optional U2U forwarding is distinct from the Overflow/Rescue workload redistribution performed in Phase 4: r_i is a task-level relay decision within the FANET, whereas Overflow and Rescue reassign pending tasks between the Mode-A and Mode-B service domains according to MEC utilization and battery conditions. The Mode-B Member-UAVs therefore complement the Head-UAV with distributed OFDMA, additional MEC capacity and flexible U2U connectivity. Together, the Mode-A Head-UAV and the Mode-B Member-UAV FANET form the heterogeneous dual-mode architecture illustrated in Figure 1.

Bidirectional Task Redistribution and Mutual Compensation. The Head-UAV's high-capacity, infrastructure-connected service is best suited for the central traffic hot spot, while the Member-UAV FANET's distributed OFDMA and U2U connectivity are best for wide-area, lower-density peripheral coverage. Neither role alone efficiently supports both regimes simultaneously, so the two domains coexist within one swarm and cooperate through bidirectional workload redistribution, without ever exchanging their structural roles—the Head-UAV remains Mode-A and the Member-UAVs remain Mode-B.

This bidirectional workload cooperation between the Mode-A Head-UAV and the Mode-B Member-UAV FANET is a key feature of A-MASH. It is controlled by the MEC utilization and the Battery Lifetime Index (BLI) and is performed in Phase 4 through two mechanisms:

- **Overflow (Mode A → Mode B):** (i) when the Head-UAV's MEC utilization exceeds 80%, $MEC_0^{util} > 0.80$, the computation of selected pending Head-UAV tasks is forwarded over U2U links to Member-UAVs with spare CPU capacity; (ii) when the battery index predicted for the end of the epoch falls below the protection threshold, $BLI_0 < 0.30$, selected Head-UAV users are handed over to Member-UAVs with a sufficient battery margin.
- **Rescue (Mode B → Mode A):** when the predicted end-of-epoch battery index of a Member-UAV u_k falls below the critical threshold, $BLI_k < 0.20$, its pending tasks and

associated vehicles are handed over to the Head-UAV, or to another Member-UAV when the Head-UAV is itself protected, thus reducing the risk of a coverage gap.

This bidirectional redistribution mitigates both computation congestion at the Head-UAV and excessive battery depletion at the Member-UAVs. Both triggers use the battery level predicted for the end of the current epoch rather than the level at its start, so that protection acts before a UAV crosses its threshold. Figure 1 illustrates the resulting architecture.

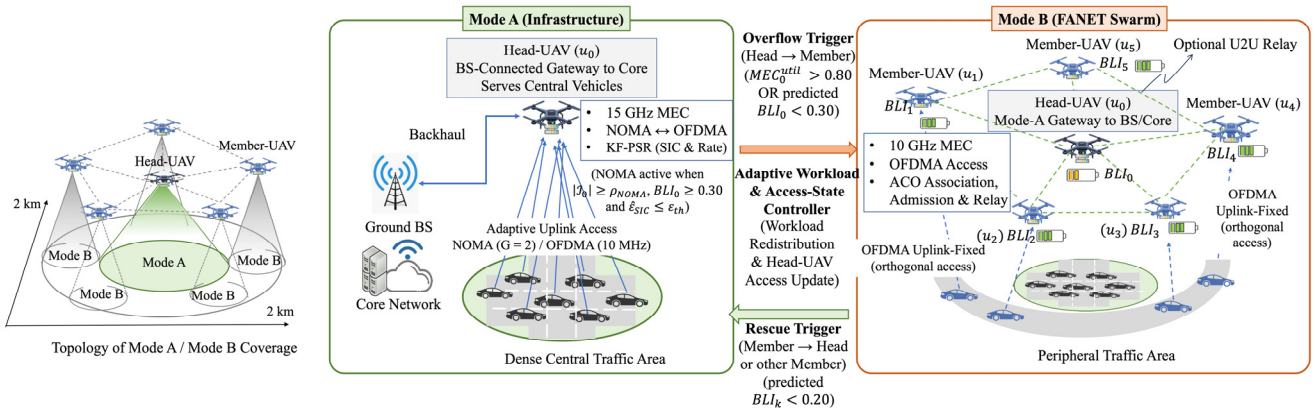


Figure 1. Conceptual architecture of the heterogeneous dual-mode UAV-IoV MEC network. The BS-connected Head-UAV u_0 (Mode A, 15 GHz MEC) serves the dense central traffic. At the end of each epoch, the Phase-4 Adaptive Workload and Access-State Controller sets its uplink access state: NOMA is activated only when the Head-UAV serves at least ρ_{NOMA} vehicles, $BLI_0 \geq 0.30$ and the estimated residual SIC factor satisfies $\hat{\varepsilon}_{SIC} \leq \varepsilon_{th}$; otherwise, it uses OFDMA. The Member-UAVs u_u (Mode B, 10 GHz MEC each) form an OFDMA FANET that serves the peripheral traffic and supports optional U2U relay forwarding. A-MASH decides vehicle association with admission control and relaying in both domains, and KF-PSR sets the SIC order and the rate targets from predicted channel gains. Overflow $(MEC_0^{util} > or BLI_0 < 0.30)$ and Rescue ($BLI_k < 0.20$) redistribute workload between the two domains, based on the battery level predicted for the end of the epoch.

Task and Decision Variables. At each scheduling epoch, each vehicle v_i, $i \in \mathcal{I}$, generates one task

$$\mathcal{T}_i = \langle D_i, C_i, \tau_i^{max} \rangle, \quad i \in \mathcal{I}, \tag{4}$$

where the input data size is $D_i \sim U[0.5, 5.0]$ Mbits for task $\mathcal{T}_i$, the computational workload at the MEC server is $C_i \sim U[500, 2000]$ Mcycles, and the maximum allowable end-to-end latency is $\tau_i^{max} = 3.0$ s (reflecting the real-time requirements of IoV applications such as collision avoidance). Since all tasks share the same deadline, we also denote it by T_{max}. Under the full-offloading assumption, the task-level decision vector for vehicle v_i is

$$\boldsymbol{s}_i = \left[a_i, f_{i,k}, \alpha_i^p, r_i\right]^T, i \in \mathcal{I}_k, k \in \mathcal{K}, \tag{5}$$

where $a_i \in \mathcal{K}$ is the serving UAV ($a_i = 0$: Head-UAV; $a_i = k \neq 0$: Member-UAV u_k), which defines the task set $\mathcal{I}_k = \{i \in \mathcal{I} \mid a_i = k\}$. The variable $f_{i,k}$ is the MEC CPU frequency allocated to task $\mathcal{T}_i$ by its serving UAV; it is optimized by GWO in Phase 1 and refined in Phase 3. The variable $\alpha_i^p \in [0.1, 1]$ is the uplink power coefficient, active only when $a_i = 0$ and $m_0 =$ NOMA ($\alpha_i^p = 1$ otherwise). The variable $r_i \in \mathcal{K} \setminus \{a_i\}$ is the optional U2U relay decision ($r_i = -1$ if unused). The association and relay variables are jointly searched by ACO in Phase 2, which evaluates candidates with the closed-form CPU split of Section 4.2; the CPU allocation is refined to $f_{i,k}^*$ in Phase 3 through the Bi-IE exchange.

Admission. When the swarm cannot serve every task within its deadline (Section 3.5), a task may also be left unadmitted, denoted $a_i = \emptyset$. It consumes no uplink or MEC resources and counts as a failed delivery, so the association variable takes values $a_i \in \mathcal{K} \cup \{\emptyset\}$.

The complete optimization vector is

$$\boldsymbol{X} = \left[\boldsymbol{s}_1^T, \ \ldots, \ \boldsymbol{s}_V^T\right]^T. \tag{6}$$

The current Head-UAV access state $m_0 \in \{NOMA, \ OFDMA\}$ is a rule-based policy state, not a GWO-ACO search variable. It is fixed during Phases 1–3 and is updated in Phase 4 according to the Head-UAV traffic density, battery condition and SIC quality. With Member-UAVs fixed to OFDMA, $\boldsymbol{X}$ has a nominal dimension of $D = 4V$ (e.g, $D = 240$ for $V = 60$). The decision space mixes continuous MEC/NOMA variables $\{f_{i,k}, \alpha_i^p\}$ with discrete association-relay variables $\{a_i, r_i\}$, which are nonlinearly coupled through the communication rates, MEC execution delay, energy consumption, and UAV battery dynamics, resulting in a high-dimensional mixed-integer nonlinear programming (MINLP) problem. A-MASH addresses this by applying GWO to the continuous resource-allocation variables $\{f_{i,k}, \alpha_i^p\}$ and ACO to the discrete variables $\{a_i, r_i\}$, subsequently couples through the Bi-IE mechanism of Section 4.

3.2. Communication Model

Air-to-Ground Path Loss. The air-to-ground propagation channel between ground vehicle v_i, $i \in \mathcal{I}$ and UAVs u_k, $k \in \mathcal{K}$, follows the standard probabilistic Line-of-Sight (LoS) and Non-Line-of-Sight (NLoS) air-to-ground model [4], with propagation characteristics considered in accordance with ITU-R P.1410 [28]. The LoS probability between vehicle v_i and UAV u_k is given by

$$P_{i,k}^{LOS} = \frac{1}{1 + a_{env} \cdot \exp[-b_{env}(\theta_{ik} - a_{env})]}, \qquad i \in \mathcal{I}, \ k \in \mathcal{K}, \tag{7}$$

where $\theta_{ik} = \tan^{-1}\left(h/d_{ik}^{hor}\right)$ is the elevation angle in degrees, h is the UAV altitude, d_{ik}^{hor} is the horizontal distance, and $(a_{env}, b_{env}) = (9.61, 0.16)$ are the environment parameters for the considered urban scenario. The average path loss combining LoS/NLoS conditions is $L_{ik}^{avg} = P_{i,k}^{LOS} \cdot L_{ik}^{FS} \cdot \eta_{LoS} + \left(1 - P_{i,k}^{LOS}\right) \cdot L_{ik}^{FS} \cdot \eta_{NLoS}$, where L_{ik}^{FS} is the free-space path loss at a carrier frequency of $f_c = 2.4$ GHz and η_{LoS}, η_{NLoS} are the corresponding attenuation factors. The channel gain is $g_{ik} = \frac{1}{L_{ik}^{avg}}$.

NOMA Uplink at the Head-UAV (Mode A, m_0 = NOMA). The $|\mathcal{I}_0|$ Head-UAV users are ranked by channel gain and interleaved into $n_c = \lceil |\mathcal{I}_0|/G \rceil$ clusters of G users, so that each cluster mixes strong and weak users; the clusters use orthogonal sub-bands of B/n_c Hz. Inside a cluster $\mathcal{C}(i)$, users are decoded in descending received power $s_j = \alpha_j^p p_{max} g_j$. A user sees the full power of the users not yet decoded and ε_{SIC} times the power of the users already cancelled, so the achievable uplink rate of vehicle v_i is

$$R_{i,0}^{NOMA} = \frac{B}{n_c} log_2\left(1 + \frac{\alpha_i^p p_{max} g_{i,0}}{\Gamma_{LA}\left(\sum_{j \in \mathcal{C}(i): s_j < s_i} \alpha_j^p p_{max} g_{j,0} + \varepsilon_{SIC} \sum_{j \in \mathcal{C}(i): s_j > s_i} \alpha_j^p p_{max} g_{j,0} + N_0 \frac{B}{n_c}\right)}\right), \ i \in \mathcal{I}_0 \tag{8}$$

where $N_0 = 10^{-20}$ W/Hz is the noise power spectral density (10^{-13} W over 10 MHz), $\Gamma_{LA} = 3$ dB is the link-adaptation margin, and $\varepsilon_{SIC} \in [0, \ 1]$ is the residual SIC factor of the receiver, identical for all users and methods. In the uplink every vehicle has its own power budget, so $\alpha_i^p p_{max}$ is simply the transmit power of vehicle v_i, with $0.1 \leq \alpha_i^p \leq 1$ and $p_{max} = 0.1$ W; no aggregate power constraint is required. The receiver performs at most $G - 1$ cancellations per cluster. Because (8) depends on the decoding order, its sum rate cannot exceed the multiple-access capacity of the cluster [29].

OFDMA Uplink (Head-UAV in OFDMA state, and all Member-UAVs). Under OFDMA, UAV u_k's bandwidth B is orthogonally shared among its $|\mathcal{I}_k|$ associated vehi-

cles, giving each vehicle $B_{i,k}^{OFDMA} = B/|\mathcal{J}_k|$; the corresponding achievable OFDMA uplink rate from vehicle v_i to UAV u_k,

$$R_{i,k}^{OFDMA} = \frac{B}{|\mathcal{J}_k|}\log_2\left(1 + \frac{p_{max}g_{ik}}{\Gamma_{LA}N_0\frac{B}{|\mathcal{J}_k|}}\right), i \in \mathcal{J}_k,\ k \in \mathcal{K}, \tag{9}$$

applied to $k = 0$ (Head-UAV, when $m_0 = OFDMA$) and to all $k \neq 0$ (Member-UAVs). Because the vehicles associated with the same OFDMA-enabled UAV are assigned mutually orthogonal bandwidth resources, intra-UAV inter-user interference is eliminated and no SIC is needed, although the channel predictor of Section 4.1.1 still sets the rate. The realized uplink rate of a Head-UAV user therefore depends on the current access state:

$$R_{i,0}^{ul} = \begin{cases} R_{i,0}^{NOMA}, & m_0 = NOMA, \\ R_{i,0}^{OFDMA}, & m_0 = OFDMA, \end{cases} \quad i \in \mathcal{J}_0. \tag{10}$$

For Member-UAVs, $R_{i,k}^{ul} = R_{i,k}^{OFDMA}, k \in \mathcal{K} \setminus \{0\}$. Thus, NOMA is used only by the Head-UAV when $m_0 = NOMA$, whereas OFDMA provides a common orthogonal-access model for the Head-UAV in its OFDMA state and for all Member-UAVs. Each UAV operates on its own 10 MHz channel, so there is no interference between UAVs, and the noise power always scales with the bandwidth allocated to a user or cluster.

UAV-to-UAV (U2U) Relay Link. A task served by UAV u_k can forward a fraction ξ_i of its input data and of its computation to another UAV u_{r_i}, $r_i \in \mathcal{K} \setminus \{k\}$, over a U2U relay link. For the optional relay selected in Phase 2, $\xi_i = \xi = 0.25$; for an Overflow forward in Phase 4 (Section 4.4), the whole computation is forwarded ($\xi_i = 1$); without forwarding, $r_i = -1$ and $\xi_i = 0$. The U2U links use a dedicated band of $B_{U2U} = 5$ MHz, a transmit power of $p_{U2U} = 0.5$ W and a line-of-sight free-space channel, because all UAVs fly at the same altitude. The U2U rate and the forwarding delay are

$$R_{k,l}^{relay} = B_{U2U}\log_2\left(1 + \frac{p_{U2U}}{L_{kl}^{FS}\eta_{LoS}N_0B_{U2U}}\right), \quad T_{i,k\to r_i}^{relay} = \begin{cases} \frac{\xi_i D_i}{R_{k,r_i}^{relay}}, & r_i \neq -1 \\ 0, & otherwise \end{cases} \tag{11}$$

where $L_{kl}^{FS} = (4\pi f_c d_{kl}/c)^2$ is the free-space path loss over the distance d_{kl} between u_k and u_l, and η_{LoS} is the LoS excess loss defined above. With the UAV positions of Section 5.1, the U2U rate is 45–54 Mbps, so forwarding $\xi D_i \leq 1.25$ Mbit takes at most 28 ms. Because the forwarded data are small, we neglect queuing on the U2U links. For a task directly served by the Head-UAV, the optional relay is not used ($r_i = -1$), since the Head-UAV already provides gateway connectivity to the BS/core network; only the Phase-4 Overflow forwards the computation of selected Head-UAV tasks ($\xi_i = 1$). As noted in Section 3.1, this task-level U2U relay is jointly optimized with a_i by ACO in Phase 2 and is distinct from the higher-level Overflow/Rescue redistribution of Phase 4.

3.3. MEC Computation Model

The serving UAV u_k allocates CPU frequency $f_{i,k}$ to task $\mathcal{T}_i$ (active only when $a_i = k$; refined to $f_{i,k}^*$ after the Bi-IE procedure of Section 4). The computation latency is $T_{i,k}^{comp} = C_i/f_{i,k}$, and the Head-UAV/Member-UAV maximum MEC capacities are $F_0^{max} = 15$ GHz and $F_k^{max} = 10$ GHz ($k \neq 0$), respectively. The normalized MEC CPU utilization and residual-capacity ratios of UAV u_k are

$$MEC_k^{util} = \frac{\sum_{i\in\mathcal{J}_k} f_{i,k}}{F_k^{max}}, \quad MEC_k^{res} = 1 - MEC_k^{util}, \quad k \in \mathcal{K}, \tag{12}$$

where a larger MEC_k^{res} indicates greater available MEC headroom at UAV u_k for newly arriving or redistributed tasks. This residual-capacity vector is passed from GWO to ACO during the Phase 3 Bi-IE refinement.

Vehicle-Side Energy Consumption. The vehicle-side energy consumption associated with task $\mathcal{T}_i$ is determined by the uplink transmission energy required to offload the task to its serving UAV u_k. Thus,

$$E_{i,k}^{veh} = p_{i,k}^{tx} T_{i,k}^{ul},\ i \in \mathcal{I}_k,\ k \in \mathcal{K}, \tag{13}$$

where $p_{i,k}^{tx}$ is the transmit power of vehicle v_i (p_{max} for OFDMA users and $\alpha_i^p p_{max}$ for Head-UAV NOMA users), and $T_{i,k}^{ul} = D_i / R_{i,k}^{ul}$ is the corresponding uplink transmission duration from Section 3.2. MEC computation energy consumed at UAV u_k is not included in $E_{i,k}^{veh}$; it is accounted for in the UAV battery model of Section 3.4. With the computation latency $T_{i,k}^{comp}$ defined in Section 3.3, the total end-to-end completion latency of task $\mathcal{T}_i$ is

$$T_{i,k}^{total} = T_{i,k}^{ul} + T_{i,k}^{proc},\ \ T_{i,k}^{proc} = \begin{cases} \frac{C_i}{f_{i,k}}, & r_i = -1 \\ \max\left\{ \frac{(1-\xi_i)C_i}{f_{i,k}}, T_{i,k\to r_i}^{relay} + \frac{\xi_i C_i}{f_{i,r_i}^{rel}} \right\}, & r_i \neq -1 \end{cases} \tag{14a}$$

where $T_{i,k}^{proc}$ is the processing time. Without a relay, the serving UAV computes the whole task. With a relay, the serving UAV computes the share $(1-\xi_i)C_i$, while the relay UAV u_{r_i} receives $\xi_i D_i$ and computes $\xi_i C_i$ in parallel; the task is complete when both parts are done. For an Overflow forward ($\xi_i = 1$), the first term is zero. The relay UAV computes forwarded shares only with its spare CPU capacity, which it divides among the forwarded tasks in proportion to their workload:

$$f_{i,l}^{rel} = min\left\{ \frac{\xi_i C_i}{\sum_{j:r_j=l} \xi_j C_j} \left(F_l^{max} - \sum\nolimits_{j \in \mathcal{I}_l} f_{j,l} \right), 0.40 F_l^{max} \right\} \tag{14b}$$

so C2 also holds at the relay UAV. A relay therefore shortens the processing time only when the relay UAV has idle CPU capacity; otherwise, it only adds delay, and the optimizer does not select it (Section 5.6). As in Section 3.1, the transfer of the small task result is neglected.

3.4. UAV Battery Model

Each UAV u_k is initialized with a battery capacity of $B_k^{init} = 50$ Wh. During a scheduling epoch of duration Δt, the energy consumption of u_k combines hovering, communication-circuit operation, MEC computation energy, and workload-dependent onboard service energy:

$$E_k^{UAV}(\Delta t) = P_{hover}\Delta t + P_{comm} T_k^{tx} + \sum_{i \in \mathcal{I}_k} \kappa_k f_{i,k}^2 (1-\xi_i) C_i \ \ + \sum\nolimits_{i:\, r_i = k} \kappa_k \left(f_{i,k}^{rel} \right)^2 \xi_i C_i + P_{dyn} |\mathcal{I}_k| \Delta t \tag{15}$$

where $P_{hover} = 100$ W is the hovering power [13], $P_{comm} = 2$ W is the receiver circuit power and T_k^{tx} is the uplink transmission time of the vehicles served by u_k, i.e., the longest uplink time among them, during which the receiver of u_k is active. The third and fourth terms are the MEC computation energy of the tasks computed on u_k: the locally computed shares of its own tasks and the shares forwarded to it as a relay. Here $\kappa_k = 10^{-28}$ J·s^2/cycle^3 is the effective switched capacitance of the MEC processor; a higher CPU frequency shortens the computation but increases its energy quadratically. The energy of the U2U transmission (0.5 W for at most about 0.1 s per forwarded task) is negligible and is not included. The last term is the workload-dependent service power per associated vehicle (link maintenance, beam tracking and control signalling). We use $P_{dyn} = 2$ W per vehicle, so that a UAV serving 40 vehicles draws 80 W in addition to hovering, comparable to the power of an embedded edge-computing module; Section 5.4 reports the sensitivity to P_{dyn} from 0 to 15 W. The scheduling epoch is $\Delta t = 30$ s. With the 50 Wh battery, hovering alone lasts 30 min, and one epoch consumes 1.7% of the battery for hovering plus 0.03% per served vehicle.

The residual battery capacity of u_k is updated as $B_k(t+\Delta t) = \left[B_k(t) - E_k^{UAV}(\Delta t)\right]^+, k \in \mathcal{K}$, where $[x]^+ = max(x, 0)$. Then, the Battery Lifetime Index (BLI) of u_k is the normalized residual battery capacity,

$$BLI_k(t) = \frac{B_k(t)}{B_k^{init}},\ 0 \le BLI_k(t) \le 1, \quad k \in \mathcal{K}. \tag{16}$$

A key motivation for the proposed framework is that minimizing total UAV energy consumption alone does not prevent premature swarm failure. In a cooperative multi-UAV system, task assignments, CPU allocation, and U2U relay routing may concentrate the workload on a small number of UAVs, causing them to deplete their batteries earlier than others. As a result, this imbalance can create local coverage holes and reduce the overall operational lifetime of the swarm. To effectively account for both total battery depletion and inter-UAV battery imbalance, we define the battery balance penalty as

$$\Gamma_{bat} = \frac{1}{|\mathcal{K}|}\sum\nolimits_{k\in\mathcal{K}}(1 - BLI_k) + \lambda_{var}\mathrm{Var}(\{BLI_k\}_{k\in\mathcal{K}}), \tag{17}$$

where the first term is the mean battery depletion of the swarm and the second term (weighted by $\lambda_{var} \ge 0$) penalizes the dispersion of the remaining battery levels. Minimizing Γ_{bat} therefore discourages concentrated computation and relay workloads. Γ_{bat} is evaluated with the BLI values predicted for the end of the current epoch; these predicted values also drive the Phase-4 Overflow/Rescue triggers and the Head-UAV access-state update (Section 3.1).

3.5. Problem Formulation

Combining the models of Sections 3.1–3.4, A-MASH jointly optimizes, at each scheduling epoch, the vehicle-to-UAV association $\{a_i\}$, MEC CPU allocation $\{f_{i,k}\}$, conditional Head-UAV NOMA power allocation $\{\alpha_i^p\}$, and optional U2U relay decisions $\{r_i\}$, with the decision vector $\mathbf{X} = \left[\mathbf{s}_1^T, \ldots, \mathbf{s}_V^T\right]^T$ as in (6). The Head-UAV access state m_0 remains the rule-based policy state of Section 3.1 and is not part of the optimization vector. The joint problem minimizes a weighted sum of five performance terms plus a deadline penalty:

$$\mathcal{P}_0 : \min\nolimits_X \mathcal{F}(\boldsymbol{X}) = w_T\nu_T\overline{T} + w_E\nu_E\overline{E} + w_Q\nu_Q\overline{Q} + w_B\nu_B\Gamma_{\mathrm{bat}} + w_S\nu_S\Omega_{\mathrm{SIC}} + \rho_{\mathrm{D}}\Phi_{\mathrm{DL}} \tag{18}$$

where $\overline{T}$ is the average completion latency, to which a rejected task contributes $2T_{max}$; $\overline{E}$ the average vehicle energy (13); $\overline{Q}$ the average MEC computation cost with $Q_{i,k} = 0.01{\cdot}f_{i,k}/F_k^{max} + 0.001{\cdot}D_i/D_{ref}$ and $D_{ref} = 10^6$ bits; and Γ_{bat} the battery-balance penalty (17). The deadline term $\Phi_{\mathrm{DL}} = (1/\mathrm{V})\sum_i[1\{\mathrm{task}\,i\,\mathrm{not\,delivered\,by}\,T_{max}\} + \lambda_i]$ uses the lateness $\lambda_i = max\left(0, T_i^{total} - T_{max}\right)/T_{max}$ for an admitted task and $\lambda_i = 1$ for a rejected one; it imposes the deadline C1 softly. The SIC-interference term, evaluated over Head-UAV NOMA users, is the mean ratio of residual to desired power:

$$\Omega_{SIC} = \begin{cases} \frac{1}{|\mathcal{J}_0|}\sum\limits_{i\in\mathcal{J}_0}\sum\limits_{j\in C(i):\ s_j>s_i} \varepsilon_{SIC}\frac{s_j}{s_i}, & m_0 = NOMA, \\ 0, & otherwise \end{cases}. \tag{19}$$

The weights $(w_T, w_E, w_Q, w_B, w_S) = (0.30, 0.25, 0.15, 0.20, 0.10)$ reflect the relative priority of latency, energy, computation cost, battery balance and SIC interference. The normalization factors $\nu_T = 1/T_{max}$ (1/s), $\nu_E = 1/(p_{max}T_{max})$ (1/J), $\nu_Q = 1/0.01$, $\nu_B = 5$ and

$\nu_S = 1$ bring the five terms to a common range, and the deadline weight is $\rho_D = 2$. The objective in (18) is subject to:

$$\begin{aligned}
&\text{C1}: \quad T_i^{\text{total}} \le \tau_i^{max},\ \forall \text{i}:\ a_i \neq \emptyset (\text{soft, through } \Phi_{\text{DL}}) \\
&\text{C2}: \quad \sum_{i \in \mathcal{I}_k} f_{i,k} \le F_k^{max},\ \forall k \in \mathcal{K}, \\
&\text{C3}: \quad 0.02 F_k^{max} \le f_{i,k} \le 0.40 F_k^{max},\ \forall i \in \mathcal{I}_k,\ \forall k \in \mathcal{K}, \\
&\text{C4a}: \quad 0.1 \le \alpha_i^p \le 1, a_i = 0, m_0 = \text{NOMA} \\
&\text{C4b}: \quad \alpha_i^p = 1, a_i \neq 0 \text{ or } m_0 = \text{OFDMA (full power)} \\
&\text{C5}: \quad a_i = \text{k} \Rightarrow BLI_k(t) \ge BLI_{min} \text{ and } \hat{BLI}_k(t + \Delta t) \ge BLI_{min}, \quad BLI_{min} = 0.20 \\
&\text{C6}: \quad a_i \in \mathcal{K} \cup \{\emptyset\},\ \forall \text{i} \in \mathcal{I} \\
&\text{C7a}: \quad r_i = -1, a_i = 0,\ \forall i \in \mathcal{I}_0, \\
&\text{C7b}: \quad r_i \in \{-1\} \cup (\mathcal{K} \setminus \{a_i\}), a_i \neq 0,\ \forall i \in \mathcal{I} \setminus \mathcal{I}_0.
\end{aligned} \tag{20}$$

Constraint C1 asks every admitted task to finish within its deadline $\tau_i^{max} = T_{max} = 3$ s. It is imposed softly through Φ_{DL} because a solution meeting it for every task does not exist at high density: with 65 GHz of MEC capacity in total and a mean workload of 1.25 Gcycles, at most about 104 tasks can meet the deadline when the uplink takes about 1 s. Tasks that cannot be served are therefore rejected ($a_i = \emptyset$, C6) instead of overloading the swarm. C2–C3 bound the per-UAV and per-task CPU allocation; together they allow at most 50 tasks per UAV, which is not binding in our scenarios. C4a–C4b bound the uplink power coefficient. C5 excludes battery-critical UAVs from new assignments and asks that the battery level predicted for the end of the epoch stays above BLI_{min}; the second part is imposed as a penalty added to (18) (10 per unit of BLI below the threshold). C7a–C7b define the relay domain. The access state m_0 is set by the rule of Section 3.1 and is not an optimization variable.

The resulting problem has nominal dimension $D = 4V$ ($D = 240$ for $V = 60$), mixing continuous variables $(f_{i,k}, \alpha_i^p)$ with discrete variables (a_i, r_i), nonlinearly coupled through access-state-dependent rates, MEC latency, energy, and battery dynamics. This high-dimensional MINLP motivates the hybrid GWO–ACO framework developed in Section 4.

4. Proposed A-MASH Framework

To solve the high-dimensional MINLP formulated in Section 3, the A-MASH framework jointly optimizes the vehicle-to-UAV association (including admission), MEC CPU-frequency allocation, conditional Head-UAV NOMA power allocation, and optional U2U relay decisions. Recall that the Head-UAV access state $m_0 \in \{NOMA,\ OFDMA\}$ is a rule-based policy state (Section 3.1), not an optimization variable. It remains fixed during Phases 1–3 and is updated only in Phase 4. Therefore, A-MASH decomposes the optimization vector $\boldsymbol{X}$ of (6) into continuous and discrete components as

$$\boldsymbol{X} = \left[\left(\boldsymbol{X}_{GWO}^{con}\right)^T,\ \left(\boldsymbol{X}_{ACO}^{disc}\right)^T \right]^T, \tag{21}$$

where $\boldsymbol{X}_{GWO}^{con}$ collects the continuous CPU and power variables and $\boldsymbol{X}_{ACO}^{disc}$ the discrete association (including admission) and U2U relay variables. A-MASH solves the problem in four coupled phases (Figure 2). Phase 1 applies GWO to the continuous variables for an initial battery-aware association, with one wolf seeded by the closed-form latency-optimal CPU split. Phase 2 applies a load- and battery-aware ACO to the discrete variables. Phase 3 couples the two searches through the Bidirectional Information Exchange (Bi-IE): the residual MEC capacity $\boldsymbol{MEC}^{res}$ of the incumbent continuous solution scales the ACO heuristic and orders guided local moves on the best discrete candidates, and the refined candidates warm-start a GWO re-allocation. Phase 4 performs Overflow/Rescue redistribution and updates m_0 for the next epoch without changing the Mode-A/Mode-B roles.

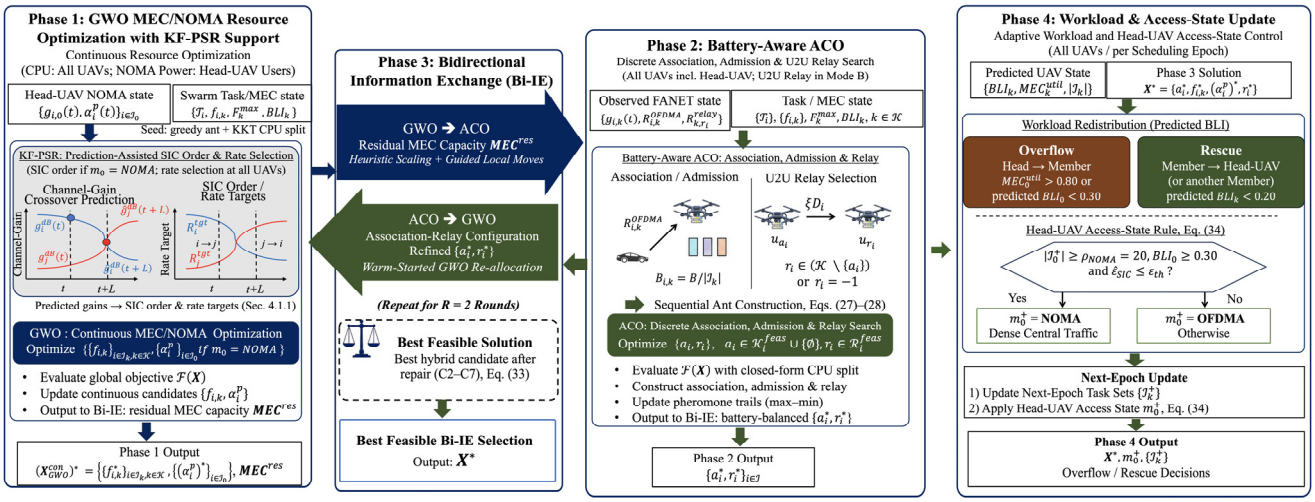


Figure 2. Four-phase A-MASH optimization procedure at one scheduling epoch. Phase 1: GWO optimizes the continuous CPU and power variables, with one wolf seeded by the closed-form CPU split, and passes the residual MEC capacity $\boldsymbol{MEC}^{res}$ to Phase 3. Phase 2: a load- and battery-aware ACO searches the discrete association, admission and U2U relay decisions. Phase 3: Bi-IE couples the two searches for $R = 2$ rounds; $\boldsymbol{MEC}^{res}$ scales the ACO heuristic and orders guided local moves (GWO → ACO), and each refined candidate warm-starts a GWO re-allocation (ACO → GWO). The best feasible hybrid solution $\boldsymbol{X}^*$ is then selected. Phases 1–3 share one budget of objective-function evaluations. Phase 4: Overflow and Rescue redistribute workload based on the battery level predicted for the end of the epoch, and the Head-UAV access m_0^+ is updated by Equation (34). KF-PSR (Section 4.1.1) sets the SIC order and the rate targets at the receiver.

4.1. Phase 1: GWO-Based Continuous MEC/NOMA Resource Optimization with KF-PSR Support

Phase 1 applies GWO to minimize the global objective $\mathcal{F}(\mathbf{X})$ in (18) over the continuous GWO decision vector

$$\boldsymbol{X}_{GWO}^{con} = \left[\{f_{i,k}\}_{i\in\mathcal{J}_k,k\in\mathcal{K}}, \{\alpha_i^p\}_{i\in\mathcal{J}_0:\, m_0=NOMA} \right]^T. \tag{22}$$

where the CPU variables are optimized for all UAVs and the power variables only for Head-UAV users when $m_0 = NOMA$; otherwise, the vehicles transmit at full power ($\alpha_i^p = 1$, C4b). The GWO population is seeded with the closed-form CPU split $f_{i,k} \propto \sqrt{C_i}$, where C_i is the workload of task i and the shares on each UAV are normalized to use its full CPU capacity. By the Karush–Kuhn–Tucker (KKT) conditions [30], this split minimizes the total computation latency of each UAV under C2. If a share violates the bounds in C3, it is clipped to C3 and the split is rescaled to satisfy C2. Of the $N = 20$ wolves, one starts at this split, nine start from slight perturbations of it (Gaussian noise with a standard deviation of 0.05 in the normalized search space), and ten start at random. The seed gives the search a feasible, latency-efficient starting point, while the random wolves keep it diverse.

4.1.1. KF-PSR: Prediction-Assisted SIC Ordering and Rate Selection

KF-PSR predicts the channel gain of each served vehicle L slots ahead (slot length $\Delta\tau = 0.1$ s), because the channel state information (CSI) available to the receiver is L slots old when the decoding order and the rate targets are set. The gain in dB is the sum of a path loss that follows the vehicle position and a log-normal shadowing term that is correlated along the trajectory with decorrelation distance d_{corr} [31]. KF-PSR therefore runs two filters per link: a constant-velocity Kalman filter [32] on the along-road position reported by the vehicle's global positioning system (GPS) receiver, and a first-order autoregressive (AR(1)) filter on the shadowing residual. At slot t they give

$$\hat{x}_i(t+L) = \hat{x}_i(t) + L\Delta\tau\hat{v}_i(t), \hat{S}_i(t) = \rho_i\hat{S}_i(t-1) + \zeta_i(t)\left[z_i(t) - PL_{dB}(\hat{x}_i(t)) - \rho_i\hat{S}_i(t-1)\right] \quad (23)$$

where $z_i(t)$ is the measured gain in dB, $\zeta_i(t)$ the scalar Kalman gain of the shadowing filter, and ρ_i the one-slot shadowing correlation computed from the estimated speed $\hat{v}_i$. The predicted gain combines the path loss at the predicted position with the decayed shadowing estimate:

$$\hat{g}_i^{dB}(t+L) = PL_{dB}(\hat{x}_i(t+L)) + \rho_i^L\hat{S}_i(t), \quad \rho_i = exp\left(-\frac{\hat{v}_i\Delta\tau}{d_{corr}}\right) \quad (24)$$

In every slot the Head-UAV orders the users of each NOMA cluster by predicted received power and sets each user's rate target from the predicted signal-to-interference-plus-noise ratio (SINR) of (8). A user is decoded only if its actual SINR reaches the target; a failed user is not cancelled and interferes fully with the users decoded after it. KF-PSR thus changes the SIC order before a predicted gain crossover and avoids rate targets that the channel will no longer support. The same predictor sets the rate of OFDMA links. Section 5.5 evaluates its accuracy and effect against a non-predictive receiver that uses the latest measurement and against perfect CSI.

4.1.2. GWO Continuous MEC/NOMA Resource Search

GWO minimizes $\mathcal{F}(\mathbf{X})$ in (18) over $\boldsymbol{X}_{GWO}^{con}$ with the standard position-update rule of the Grey Wolf Optimizer [10]: the three fittest candidates (α, β, δ wolves) guide the population, and the exploration–exploitation parameter decreases linearly over the iterations. The power dimensions are active only for Head-UAV users when $m_0 = NOMA$.

After each GWO update, the candidates are projected onto their bounds—$f_{i,k}$ onto C2–C3 and α_i^p onto C4—and the remaining constraints are enforced by the common repair step of Section 4.3. After the maximum number of iterations, the best solution is

$$(\boldsymbol{X}_{GWO}^{con})^* = \left[\left\{f_{i,k}^*\right\}_{i\in\mathcal{I}_k, k\in\mathcal{K}}, \left\{(\alpha_i^p)^*\right\}_{i\in\mathcal{I}_0:\, m_0=NOMA}\right]^T. \quad (25)$$

Based on the optimized CPU-frequency allocation, the MEC CPU utilization and residual MEC-capacity ratio of each UAV are calculated according to (12). The resulting residual MEC-capacity vector is $\boldsymbol{MEC}^{res} = \left[MEC_0^{res}, \ldots, MEC_{K-1}^{res}\right]^T$, which is passed to the Bi-IE stage of Phase 3 as GWO-to-ACO information that guides the discrete association and relay search.

4.2. Phase 2: Battery-Aware ACO for Discrete Association, Admission and U2U Relay Search

Given the continuous resource state obtained in Phase 1, Phase 2 applies ACO to jointly search the discrete vehicle-to-UAV association a_i and optional U2U relay decision r_i. The serving-UAV association is selected from the feasible UAV set including the Head-UAV and Member-UAVs, while U2U relay selection is available only for tasks served by Member-UAVs. The ACO decision vector is defined as

$$\boldsymbol{X}_{ACO}^{disc} = \left[\{a_i\}_{i\in\mathcal{I}}, \{r_i\}_{i\in\mathcal{I}}\right]^T, \quad (26)$$

where $a_i \in \mathcal{K} \cup \{\emptyset\}$ is the serving UAV of task $\mathcal{T}_i$ or the rejection option. ACO does not optimize $f_{i,k}$ or α_i^p directly: every ant solution is evaluated with the closed-form CPU split of Section 4.1 and the incumbent power coefficients, and the continuous block is re-optimized in Phase 3. Battery-critical UAVs are excluded from the candidate set $\mathcal{K}_i^{feas} = \{k \in \mathcal{K} \mid BLI_k \geq BLI_{min}\}$ (C5).

Ants assign the vehicles one at a time in a random order, so that the heuristic accounts for the load already placed on each UAV. When vehicle v_i is considered, the CPU share it

would receive on UAV u_k, given the set $\mathcal{J}_k^{cur}$ of tasks already assigned there, is estimated by the closed-form split

$$\hat{f}_{\mathrm{i,k}} = \min\left\{\max\left\{\frac{F_k^{max}\sqrt{C_i}}{\sum_{j\in\mathcal{J}_k^{cur}}\sqrt{C_j}+\sqrt{C_i}},\ 0.02F_k^{max}\right\},\ 0.40F_k^{max}\right\}, k \in \mathcal{K}_i^{feas} \tag{27}$$

and $T_{i,k}^{ul}$ is the uplink delay estimated with the current number of vehicles on u_k plus one. The resulting completion-time estimate $\hat{T}_{i,k} = T_{i,k}^{ul} + C_i/\hat{f}_{i,k}$ defines the battery-aware heuristic of the association and admission options:

$$\eta_{i,k} = \frac{BLI_k}{\hat{T}_{i,k}} \cdot min\left\{1,\ \left(\frac{T_{max}}{\hat{T}_{i,k}}\right)^2\right\}, \quad \eta_{i,\emptyset} = \frac{\overline{BLI}}{2T_{max}} \tag{28}$$

The heuristic favours UAVs with a high remaining battery and a short estimated completion time, and strongly discounts UAVs on which the task would miss its deadline; the rejection option has a weak constant heuristic, where $\overline{BLI}$ is the mean BLI of the feasible UAVs. An analogous relay heuristic $\eta_{i,k,l}^{relay} = BLI_l/\left(T_{i,k\to l}^{relay} + \delta_0\right)$, with $\delta_0 = 1$ ms, is used for candidate relay UAVs u_l, $l \in \mathcal{K} \setminus \{k\}$, only when $a_i = k \neq 0$ (C7a). Ants choose with the pheromone-guided transition probability [33] proportional to $\tau_{i,k} \cdot \eta_{i,k}^{\beta}$ with $\beta = 2$. Pheromone is updated in max–min form: all trails evaporate at rate 0.1, the iteration-best and best-so-far solutions deposit $1/\mathcal{F}(\mathbf{X})$, and trails are bounded to $[0.1, 10]$ to avoid stagnation. After the Phase-2 iterations, ACO returns its best discrete candidates

$$\left(\boldsymbol{X}_{ACO}^{disc}\right)^* = \left[\ \{a_i^*\}_{i\in\mathcal{I}}\ ,\ \{r_i^*\}_{i\in\mathcal{I}}\ \right]^T. \tag{29}$$

The residual MEC capacity $\boldsymbol{MEC}^{res}$ is deliberately not used in Phase 2; it enters only through Bi-IE in Phase 3.

4.3. Phase 3: Bidirectional Information Exchange (Bi-IE) and Best-Feasible Selection

Phase 1 provides the continuous GWO solution $(\boldsymbol{X}_{GWO}^{con})^*$ and the residual MEC CPU capacity vector $\boldsymbol{MEC}^{res}$, while Phase 2 provides the discrete solution $\left(\boldsymbol{X}_{ACO}^{disc}\right)^*$. However, these solutions cannot be directly concatenated, because a change in a_i changes the task set $\mathcal{J}_k$ of each UAV and hence its MEC CPU allocation. To resolve this, A-MASH uses the Bi-IE mechanism, exchanging information in both directions while m_0 remains fixed:

GWO → ACO. In Phase 3, the residual MEC-capacity information obtained from GWO is additionally incorporated into the association as

$$\tilde{\eta}_{i,k}^{assoc} = \eta_{i,k}^{assoc} \cdot max(MEC_k^{res}, 0.05), \tag{30}$$

so that congested UAVs are penalized even if they are otherwise battery- and delay-favourable. ACO then generates new candidates with the scaled heuristic. In addition, $\boldsymbol{MEC}^{res}$ orders a guided local-move step applied first to the incumbent and then to the best new candidates: tasks on the most loaded UAVs and tasks that miss their deadline are visited first, and each is moved to the UAV, or to the rejection option, that lowers $\mathcal{F}(\mathbf{X})$ most. Sweeps repeat until none improves or the round's budget is spent.

ACO → GWO. For each refined discrete candidate $\left(\boldsymbol{X}_{ACO}^{disc}\right)^{(cand)}$, the association-dependent task sets $\mathcal{J}_k^{(cand)}$ are reconstructed, and GWO performs a limited, warm-started refinement of the continuous allocation,

$$\left((\boldsymbol{X}_{GWO}^{con})^*\right)^{(cand)} = \underset{\boldsymbol{X}^{cont}}{argmin}\,\mathcal{F}\left(\boldsymbol{X}^{cont}, \left(\boldsymbol{X}_{ACO}^{disc}\right)^{(cand)};\ m_0\ \right), \tag{31}$$

which is solved under the fixed discrete candidate and the fixed access state m_0. The warm start places one wolf at the closed-form CPU split of the candidate and keeps the incumbent

power coefficients; half of the population perturbs this point slightly. The final $f^*_{i,k}$ is therefore determined only after the association structure has been refined.

When $m_0 = NOMA$, the power coefficients of the Head-UAV users are re-optimized in the same warm-started GWO run. The resulting hybrid candidate

$$\boldsymbol{X}_{hyb}^{(cand)} = \left[\left(\left((\boldsymbol{X}_{GWO}^{con})^* \right)^{(cand)} \right)^T, \left(\left(\boldsymbol{X}_{ACO}^{disc} \right)^{(cand)} \right)^T \right]^T. \tag{32}$$

is evaluated via $\mathcal{F}(\mathbf{X})$ under all constraints, and the best feasible hybrid solution is selected as

$$\boldsymbol{X}^*_{Bi-IE} = arg \min_{\boldsymbol{X}_{hyb} \in \mathcal{X}_{feas}} \mathcal{F}\left(\boldsymbol{X}_{hyb}\right), \tag{33}$$

where $\mathcal{X}_{feas}$ is the set of hybrid candidates that satisfy C2–C7 after repair. The same repair is applied to every method in Section 5: an association to a battery-critical UAV is moved to the feasible UAV with the largest channel gain; relays are removed for Head-served tasks and for battery-critical relay UAVs; CPU allocations are clipped to C3 and scaled down proportionally per UAV to meet C2; and power coefficients are clipped to C4. Bi-IE repeats the two exchanges for $R = 2$ rounds. All evaluations of Phases 1–3 count against one budget of objective-function evaluations (6000 in Section 5), split 10/20/70% between Phases 1, 2 and 3; within a round, 15% goes to ACO, 60% to the guided local moves and 25% to GWO.

4.4. Phase 4: Adaptive Workload Redistribution and Head-UAV Access-State Update

Phase 4 performs system-level workload redistribution and updates the Head-UAV access state for the subsequent scheduling epoch. Unlike Phases 1–3, this phase does not introduce an additional GWO or ACO search variable. Instead, it uses the Phase 3 solution, the resulting MEC utilization, and the BLI values predicted for the end of the epoch to determine whether workload redistribution is required and to update the Head-UAV access state.

First, Overflow is evaluated with the state predicted for the end of the epoch. If $MEC_0^{util} > 0.80$, the computation of the Head-UAV tasks with the largest CPU share is forwarded over U2U links to the Member-UAVs with the most spare CPU capacity, among those whose predicted BLI is at least 0.30, until the utilization falls to 0.80. If the predicted $BLI_0 < 0.30$, the Head-UAV users with the best alternative links are handed over to Member-UAVs whose predicted BLI stays above 0.30. Second, Rescue is evaluated for every Member-UAV u_k: if its predicted $BLI_k < 0.20$, its pending tasks are handed over to the Head-UAV. If the Head-UAV is itself protected (predicted $BLI_0 < 0.30$), they are handed over to the Member-UAV with the best link among those whose predicted BLI is at least 0.20. The CPU split of every UAV affected by these moves is then recomputed.

The thresholds follow from the battery and CPU budgets. $BLI_{min} = 0.20$ keeps 10 Wh of the 50 Wh battery, about 6 min of hovering at 100 W, as a reserve for the return flight and landing; a UAV that reaches it is withdrawn (Section 5.4). The protection threshold 0.30 lies 0.10 above this reserve. This margin equals about 3 min of hovering, or six epochs, so Overflow can hand the Head-UAV users over gradually before the Head-UAV must leave. The utilization threshold 0.80 leaves 20% of the Head-UAV CPU (3 GHz) as headroom. Section 5.4 shows that the results are insensitive to the last two values.

After the Overflow/Rescue operations, the next-epoch Head-UAV access state is updated according to

$$m_0^+ = \begin{cases} NOMA, & |\mathcal{J}_0^+| \geq \rho_{NOMA} \ \wedge \ BLI_0^+ \geq 0.30 \ \wedge \ \hat{\varepsilon}_{SIC} \leq \varepsilon_{th}, \\ OFDMA, & otherwise. \end{cases} \tag{34}$$

where $\mathcal{J}_0^+$ denotes the Head-UAV association set prepared for the subsequent scheduling epoch, and BLI_0^+ is the BLI of the Head-UAV at the start of that epoch. The Member-UAV

access states remain fixed as $m_k = OFDMA, k \in \mathcal{K} \setminus \{0\}$. Thus, Phase 4 prepares the workload and access state for the next scheduling epoch without altering the structural Mode-A/Mode-B roles or the dimensionality of the optimization problem. Because Phase 4 is a system-level controller, Section 5 applies it in the same way to the solution of every optimization method, so that the comparison isolates the optimizer.

The Complete A-MASH Algorithm. Algorithm 1 summarizes the full four-phase A-MASH procedure, jointly optimizing the continuous CPU and power resources and the discrete association, admission and relay decisions through GWO, ACO and Bi-IE, and then performs Overflow/Rescue workload control and the Head-UAV access state update. Its outputs are the best feasible solution $\boldsymbol{X}^*$, the next-epoch Head-UAV state m_0^+, and the updated task sets $\{\mathcal{I}_k^+\}$.

Algorithm 1. Complete A-MASH framework with mobility-robust NOMA support and Adaptive Workload/Access-State Control

Require: tasks $\{\widetilde{\mathcal{I}}_i\}$, UAV set $\mathcal{U}$, $\{BLI_k\}$, m_0, gains $\{g_{i,k}\}$, budget N_{FE}, population N, rounds R
Ensure: $\boldsymbol{X}^*$, m_0^+, $\{\mathcal{I}_k^+\}$

// [Phase 1] GWO on the continuous block
Build an initial association with one greedy ant (Equations (27) and (28)); seed one wolf with the closed-form CPU split
Optimize $\boldsymbol{X}_{GWO}^{con}$ in Equation (22) using $\mathcal{F}(\mathbf{X})$ in Equation (18)

// [Phase 2] Battery-aware ACO on the discrete block
for each ACO iteration **do**
construct N ants sequentially with Equations (27) and (28); evaluate; update pheromone
end for

// [Phase 3] Bidirectional Information Exchange (Bi-IE)
for $r = 1, \ldots, R$ do
GWO → ACO: compute $\boldsymbol{MEC}^{res}$ by Equation (12);
Scale the heuristic by Equation (30); Run ACO
Apply $\boldsymbol{MEC}^{res}$-guided local moves to the incumbent and the best candidates
ACO → GWO: warm-started GWO re-allocation of each refined candidate, Equation (31)
end for
Select the best feasible hybrid solution $\boldsymbol{X}^*$ by Equation (33)

// [Phase 4] System-level workload and access-state control
Apply Overflow if $MEC_0^{util} > 0.80$ or $\hat{BLI_0} < 0.30$;
Apply Rescue if $\hat{BLI}_k < 0.20$, $k \neq 0$
Update $\{\mathcal{I}_k^+\}$ and m_0^+ by Equation (34)
Return $\boldsymbol{X}^*$, m_0^+, $\{\mathcal{I}_k^+\}$

4.5. Computational Complexity

One evaluation of $\mathcal{F}(\mathbf{X})$ costs $O(VK)$ for the association-dependent rates, CPU sums and battery terms, plus $O(G|\mathcal{I}_0|)$ for the NOMA clusters, i.e., $O(V(K+G))$. An ant construction costs $O(VK)$, and the closed-form CPU split costs $O(V)$ when no bound

of C3 is active and at most $O(V^2/K)$ otherwise. Because every evaluation of Phases 1–3 counts against the budget $N_{\mathbf{FE}}$ of objective-function evaluations (FEs), the run time of A-MASH is $O(N_{FE} \cdot V(K+G))$ plus the CPU-split overhead, the same order as a standalone metaheuristic with the same budget. One sweep of the alternating-optimization baseline AO-BCD of Section 5 evaluates $V(K+1)$ moves, each with a CPU re-split, i.e., $O(V^2K(K+G))$. Measured run times are reported in Section 5.7: with $N_{FE} = 6000$, a decision takes at most about 2.2 s on one CPU core, about 7% of the 30 s scheduling epoch.

5. Simulation and Results

5.1. Simulation Setup

We evaluate A-MASH with a Python 3.13 simulator written for this study. The service area is 2 km × 2 km with a Manhattan grid of roads at $x, y \in \{400, 1000, 1600\}$ m. Half of the vehicles drive on the two central roads within a 1 km downtown segment around the centre, which creates the dense central traffic served by the Head-UAV; the others drive on the four peripheral roads. Each vehicle starts at a uniform position on its road, moves in either direction at a speed drawn uniformly from 60 to 120 km/h, and wraps around its segment so that the density stays constant. The Head-UAV hovers at (1000, 1000, 100) m and the five Member-UAVs on a ring of radius 650 m at azimuths $90° + 72°(k-1)$, $k = 1, \ldots, 5$.

Each vehicle generates one task per 30 s scheduling epoch. A single-epoch evaluation starts from a mid-mission battery state, $BLI_k \sim U[0.4, 1]$, and the optimized solution is then executed by a slot-level simulator (slot 0.1 s, horizon 10 s): vehicles move, shadowing evolves, the receiver uses gain measurements that are $L = 2$ slots old with 1 dB error, and a user is decoded only if its actual SINR reaches its rate target (Section 4.1.1). All reported latencies, PDRs, energies and throughputs come from this execution. The lifetime study of Section 5.4 instead starts from full batteries and runs 50 consecutive epochs (25 min).

For each density $V \in \{10, 20, 40, 60, 80, 100, 120, 150\}$ we run 30 independent scenarios, which differ in vehicle positions and speeds, tasks, shadowing and initial battery states; every method is evaluated on the same 30 scenarios. The seeds used for development are disjoint from those used for the reported results. $U[a, b]$ denotes a uniform distribution. Results are given as mean ± 95% confidence interval, computed as mean $\pm\, t_{0.975,29} \cdot s/\sqrt{30}$. Table 3 lists all parameters.

Table 3. Simulation parameters.

Parameter	Value	Parameter	Value
Service area	2 km × 2 km	UAV altitude h	100 m
UAVs K	6 (1 Head + 5 Members)	Vehicles V	10–150
CPU F_0^{max}/F_k^{max}	15 GHz/10 GHz	CPU bounds (C3)	0.02–0.40 F_k^{max}
Task input D_i	$U[0.5, 5]$ Mbit	Task workload C_i	$U[500, 2000]$ Mcycles
Deadline T_{max}	3 s	Bandwidth B	10 MHz per UAV
Noise N_0	10^{-20} W/Hz	Vehicle power p_{max} oefficient α_i^p	0.1 W; [0.1, 1]
Path loss $(a,\ b,\ \eta_{LoS},\ \eta_{NLoS})$	9.61, 0.16, 1 dB, 20 dB	Carrier f_c	2.4 GHz
Shadowing σ, d_{corr}	4 dB, 50 m	Link margin Γ_{LA}	3 dB
NOMA cluster G, ε_{SIC}	2, 0.05	ρ_{NOMA}, ε_{th}	20, 0.0025
Slot $\Delta\tau$, CSI delay L	0.1 s, 2 slots (0.2 s)	Gain/GPS error (std.)	1 dB/3 m

Table 3. *Cont.*

Parameter	Value	Parameter	Value
U2U band, power	5 MHz, 0.5 W	Relay fraction ξ	0.25
Battery B_k^{init}	50 Wh	P_{hover}, P_{comm}	100 W, 2 W
Switched capacitance κ	10^{-28} J·s^2/cycle^3	$P_{\mathbf{dyn}}$	2 W per vehicle
Epoch Δt	30 s	Initial BLI	$U[0.4,\ 1]$ (lifetime: 1)
BLI_{min}; Overflow thresholds	0.20; 0.80 (utilization), 0.30 (BLI)	Weights $(w_T, w_E, w_Q, w_B, w_S)$	0.30, 0.25, 0.15, 0.20, 0.10
λ_{var}, ρ_D	0.5, 2	ν_T, ν_E, ν_Q, ν_B, ν_S	1/3 s^{-1}, 1/0.3 J^{-1}, 100, 5, 1
Evaluation budget N_{FE}	6000 per decision	Population/ants	30 (A-MASH: 20)
Runs	30 per density	Bi-IE rounds R	2

5.2. Baselines and Evaluation Metrics

We compare A-MASH with seven methods. Standalone GWO [10], PSO [34] (inertia 0.7, acceleration coefficients 1.5 and 1.5), continuous ACO ($ACO_{\mathbf{R}}$) [11] and SMA [35] search the whole decision vector through a continuous encoding of association (including rejection), CPU share, power coefficient and relay; Random samples the same encoding uniformly. Greedy associates each vehicle with the feasible UAV of largest channel gain, splits CPU with the closed-form rule and, while an admitted task would miss its deadline, rejects the latest task of each such UAV. Alternating optimization with block-coordinate descent (AO-BCD) starts from Greedy and alternates the closed-form CPU block, a per-vehicle best-response block over {feasible UAVs, reject} and a power grid block until it reaches a fixed point.

All methods receive the same budget of 6000 objective-function evaluations (FE) per decision, counting every internal evaluation; AO-BCD stops earlier when it reaches a fixed point. They share the scenarios, the repair step of Section 4.3, the Phase-4 controller and the slot-level execution, so that only the optimizer differs.

The packet (task) delivery ratio (PDR) is the share of all generated tasks delivered within T_{max}, so rejected tasks, failed uploads and late tasks all count as failures. The deadline-violation probability is the share of admitted tasks finished after T_{max}. Latency statistics are computed over admitted tasks; an upload not finished within 10 s counts at 10 s. Vehicle energy is the mean transmit energy per generated task (zero for a rejected task), and throughput is the sum of the achieved per-user uplink rates. Γ_{bat} and BLI_{min} are evaluated at the end of the epoch. Relative differences are reported per density and per baseline, with paired Wilcoxon signed-rank tests over the 30 scenarios and Holm correction across baselines.

5.3. Comparison with the Baselines

Figure 3 compares the eight methods over the density range. Up to $V = 40$, A-MASH, AO-BCD and Greedy deliver nearly every task. From $V = 60$ on, the standalone metaheuristics and Random fall behind sharply: at $V = 100$ their PDR is 0.38–0.53 against 0.871 for A-MASH and their 95th-percentile latency exceeds 5 s. A-MASH also needs 22–37% less vehicle energy than each of them except $ACO_{\mathbf{R}}$, which saves energy by rejecting 30% of the tasks. A search over a $4V$-dimensional encoding of mixed variables cannot exploit the problem structure within the budget. Table 4 summarizes the relative differences with their significance.

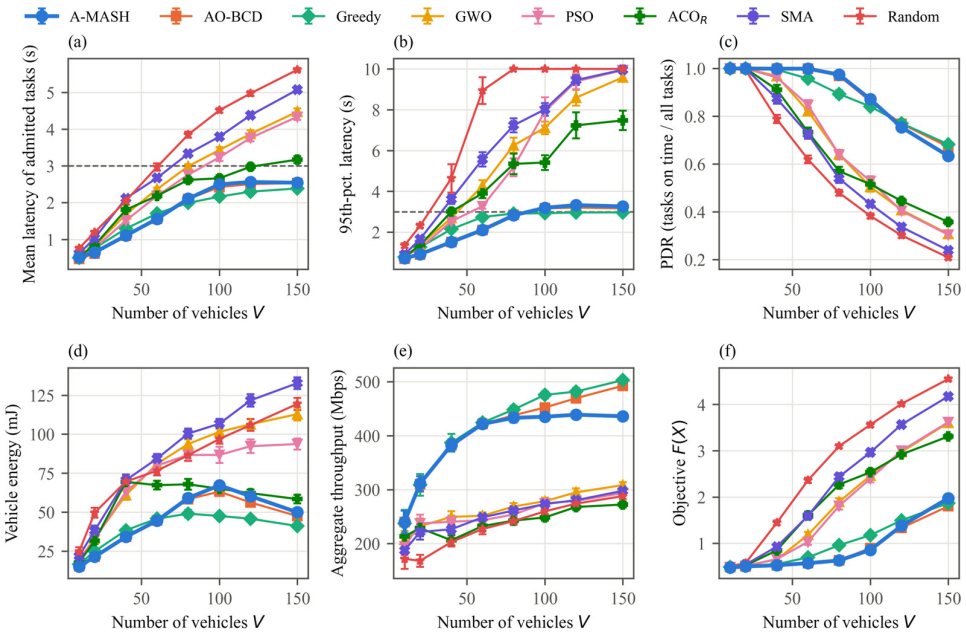


Figure 3. Comparison of A-MASH with the baselines versus the number of vehicles: (**a**) mean latency of admitted tasks, (**b**) 95th-percentile latency, (**c**) PDR, (**d**) vehicle energy, (**e**) aggregate throughput and (**f**) objective value. The dashed lines mark the 3 s deadline. Markers are means over 30 scenarios; error bars are 95% confidence intervals.

Table 4. Relative difference of A-MASH to each baseline: PDR (higher is better)/objective value (positive = A-MASH lower), in %. * marks a significant difference (paired Wilcoxon test, Holm-corrected $p < 0.05$, 30 scenarios).

V	AO-BCD	Greedy	GWO	PSO	ACO$_R$	SMA	Random
40	+0.0/+0.0	+0.4/+4.4 *	+3.2 */+18.9 *	+3.8 */+20.1 *	+9.6 */+37.9 *	+14.3 */+43.3 *	+26.7 */+63.5 *
60	+0.1/+0.0	+4.4 */+17.9 *	+21.7 */+51.7 *	+17.9 */+44.7 *	+36.2 */+64.4 *	+37.6 */+64.1 *	+61.3 */+75.8 *
80	+0.5/+0.4	+9.0 */+34.4 *	+52.0 */+66.8 *	+52.4 */+65.2 *	+70.9 */+72.3 *	+80.7 */+74.2 *	+103.1 */+79.7 *
100	+1.0/+4.2 *	+3.6 */+27.0 *	+72.7 */+65.1 *	+64.9 */+64.2 *	+69.1 */+66.2 *	+101.5 */+71.0 *	+127.7 */+75.9 *
120	−1.9 */−3.5 *	−2.2 */+7.8 *	+85.0 */+53.6 *	+86.8 */+54.0 *	+69.3 */+52.8 *	+124.3 */+61.3 *	+150.4 */+65.6 *
150	−5.7 */−9.5 *	−7.1 */−5.5 *	+106.3 */+45.2 *	+108.4 */+45.4 *	+77.4 */+40.3 *	+165.5 */+52.7 *	+203.6 */+56.6 *

The two model-based baselines are much stronger. A-MASH and AO-BCD perform alike up to $V = 80$. At $V = 100$, A-MASH reaches a 4.2% lower objective ($p = 0.001$) with a similar PDR (0.871 vs. 0.863, not significant). At $V = 120$ and 150, above the capacity bound of Section 3.5, AO-BCD and even Greedy are better: the PDR of A-MASH is 1.9% and 5.7% below AO-BCD, and 2.2% and 7.1% below Greedy. In this regime most decisions are admission decisions, and the deterministic rejection rule shared by both baselines is an efficient heuristic that A-MASH has to rediscover by search; Section 5.7 shows that a larger evaluation budget narrows this gap but does not close it.

Figure 4 shows how the methods use the swarm. The model-based methods admit every task up to $V = 80$; at $V = 150$, A-MASH admits 71% of the tasks, AO-BCD 75% and Greedy 72% (Figure 4a). Among the admitted tasks, A-MASH and AO-BCD miss the deadline in about 10–13% of the cases from $V = 100$ on, Greedy in 3.7–4.8% because it

admits fewer tasks, and GWO in up to 63% (Figure 4b). As the density grows, A-MASH moves load away from the Head-UAV more than the other methods: at $V = 100 - 150$ the Head-UAV serves 17–22% of the vehicles under A-MASH against 19–25% under Greedy (Figure 4c). Within one 30 s epoch the battery metrics hardly differ between the methods ($|\Delta\Gamma_{bat}| \leq 0.002$ and $|\Delta BLI_{min}| \leq 0.002$ between A-MASH and any baseline), because an epoch consumes 1.7% of the battery for hovering and only a few hundredths of a percent per served vehicle; battery-aware behaviour is therefore evaluated over full missions in Section 5.4.

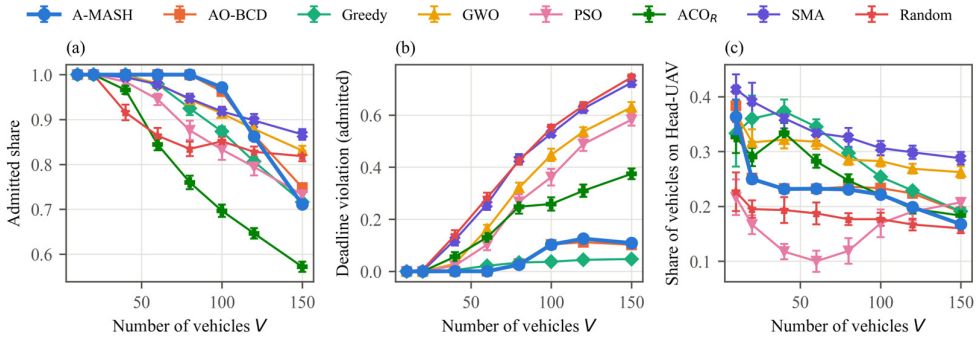


Figure 4. (**a**) Share of admitted tasks, (**b**) deadline-violation probability of admitted tasks and (**c**) share of vehicles served by the Head-UAV versus the number of vehicles (30 scenarios; error bars are 95% confidence intervals).

Figure 5 and Table 5 give the latency distribution, with its median (P50) and its 95th and 99th percentiles (P95, P99). A-MASH and AO-BCD keep the 95th percentile within about 0.3 s of the deadline at every density, while the standalone metaheuristics show heavy tails. Greedy reaches lower percentiles because it admits fewer tasks, which costs it PDR up to $V = 100$. Above the capacity bound the deadline cannot be guaranteed for every admitted task: about 4% of the admitted tasks are planned as late by the optimizer, because rejecting them would raise the objective more, and the remaining misses arise during execution from channel variation and outages.

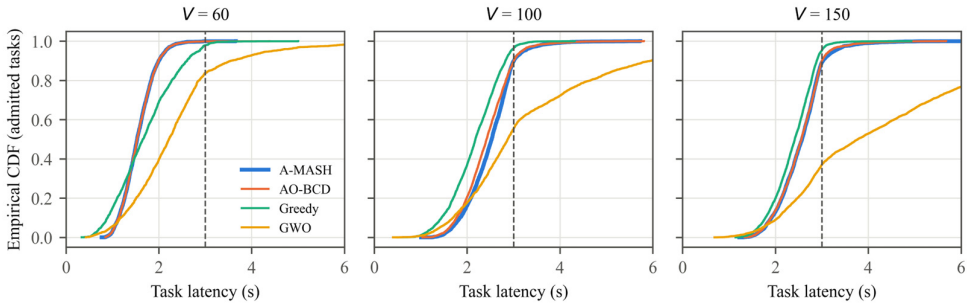


Figure 5. Empirical CDF of the latency of admitted tasks at $V = 60$, 100 and 150 (all tasks of the 30 scenarios pooled). The dashed line marks the 3 s deadline.

Table 5. Latency percentiles and deadline-violation probability of admitted tasks.

V	Method	P50 (s)	P95 (s)	P99 (s)	Deadline Violation
60	A-MASH	1.53	2.14	2.48	0.1%
60	AO-BCD	1.53	2.15	2.43	0.1%
60	Greedy	1.66	2.87	3.11	2.2%
60	GWO	2.22	4.44	6.78	16.3%

Table 5. *Cont.*

V	Method	P50 (s)	P95 (s)	P99 (s)	Deadline Violation
100	A-MASH	2.53	3.21	3.91	10.3%
100	AO-BCD	2.42	3.19	3.91	10.4%
100	Greedy	2.18	2.94	3.25	3.7%
100	GWO	2.87	7.49	10.00	44.8%
150	A-MASH	2.57	3.27	3.96	10.9%
150	AO-BCD	2.56	3.17	3.90	10.2%
150	Greedy	2.42	2.99	3.25	4.8%
150	GWO	3.78	10.00	10.00	63.0%

5.4. Battery Behaviour over Full Missions

The single-epoch results cannot show the effect of the battery-aware mechanisms, so we simulate 25 min missions (50 epochs) from full batteries at $V = 60$, 100 and 150, with ten scenarios each. Vehicles keep moving and generate new tasks in every epoch. When the BLI of a UAV reaches $BLI_{min} = 0.20$ at the end of an epoch, it returns to base for the rest of the mission, and its vehicles are served by the remaining UAVs; once all UAVs have left, the remaining tasks fail.

Figure 6 shows the number of UAVs still in service. With $P_{dyn} = 2$ W, hovering dominates the consumption, so every UAV leaves between about 15 and 21 min whichever optimizer is used. A-MASH keeps all six UAVs in service the longest: its first withdrawal occurs after 19.5, 17.5 and 17.0 min at $V = 60$, 100 and 150, one epoch (0.5 min) later than without the Overflow/Rescue controller (paired Wilcoxon $p = 0.002$), 0.45–0.55 min later than with AO-BCD at $V = 100$ and 150 ($p < 0.005$; equal at $V = 60$), and 0.5–1.9 min later than Greedy and GWO. Because A-MASH balances the batteries, its UAVs also leave closer together: the last one leaves 0.4–1.0 min earlier than with Greedy.

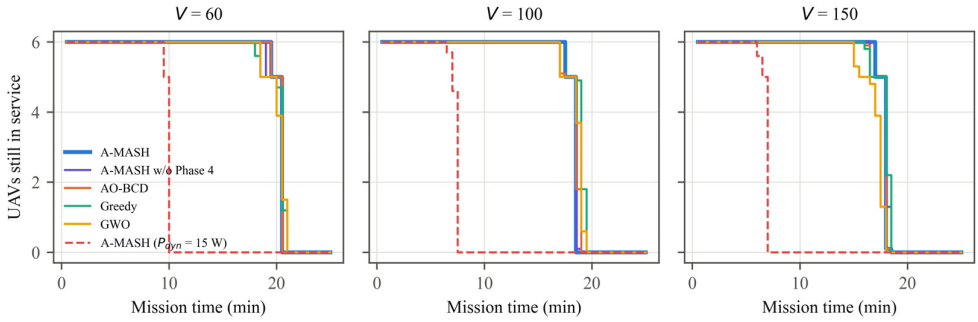


Figure 6. Number of UAVs still in service during 25 min missions starting from full batteries (mean of ten scenarios) at $V = 60$, 100 and 150. A UAV is withdrawn when its BLI reaches 0.20.

A-MASH also avoids emergency handovers. The Rescue rule moves no vehicle in any A-MASH mission at $V = 60$ and 100 and in nine of the ten missions at $V = 150$. In contrast, it moves 47–50 vehicles per mission on average under Greedy, 46–115 under GWO and up to 52 under AO-BCD. The controller has a cost: it hands Head-UAV users over to Member-UAVs with weaker links, which lowers the mission PDR by 2.4–2.6 percentage points at $V = 100$ and 150 ($p = 0.002$). In summary, the battery-aware mechanisms postpone the first loss of coverage by about one scheduling epoch. The gain is small because hovering dominates the energy budget, and Phase 4 trades some reliability for this protection.

We repeated the missions with the Overflow utilization threshold set to 0.70 and 0.90 instead of 0.80, and with the protection threshold set to 0.25 and 0.35 instead of 0.30, one at a time and with the same ten missions per density. The utilization threshold has no effect, because during the missions the Member-UAVs never have spare CPU capacity

to take over Head-UAV computation. Changing the protection threshold moves the first withdrawal by at most 21 s on average, less than one epoch, and the time until all UAVs are withdrawn by at most 3 s; the mission PDR changes by at most 0.6 percentage points. The only significant changes occur at $V = 100$ with the threshold 0.35, which leads to fewer handovers (47 instead of 63 per mission): the first withdrawal comes 21 s earlier ($p = 0.016$) and the mission PDR rises by 0.6 points ($p = 0.002$), the same trade-off between protection and PDR as above. We did not vary $BLI_{min} = 0.20$, because it also defines when a UAV must return.

Figure 7 varies the per-vehicle service power at $V = 100$. The first withdrawal moves from 24.0 min without service power (hovering only, until $BLI = 0.20$) to 17.5 min at 2 W, 12.5 min at 5 W and 6.9 min at 15 W, and the mission PDR falls from 0.835 to 0.231. With 15 W per vehicle, a UAV serving 40 vehicles would draw 600 W in addition to hovering. For this reason, we use a much lower value in this paper. The absolute mission time depends strongly on this parameter, so it should be calibrated to the actual MEC payload.

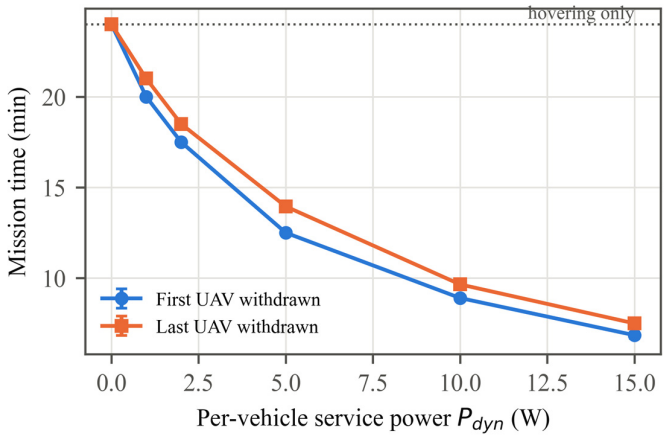


Figure 7. Mission time until the first and the last UAV withdrawal versus the per-vehicle service power P_{dyn} (A-MASH, $V = 100$, mean and 95% confidence interval over ten scenarios). Most confidence intervals are zero, because all ten missions withdraw the UAV in the same 30 s epoch; the others are at most ±0.17 min and are hidden by the markers. The dotted line marks the 24 min that hovering alone allows before BLI reaches 0.20.

5.5. Prediction-Assisted SIC and NOMA Operation

To evaluate KF-PSR, each A-MASH solution with the Head-UAV forced into NOMA is executed three times on identical channel realizations: with a non-predictive receiver that uses the latest measurement, with KF-PSR, and with perfect CSI. Figure 8 reports the results at $V = 100$ over vehicle speeds of 60, 90 and 120 km/h and CSI delays of 1, 2 and 5 slots.

KF-PSR lowers the gain-prediction error in every case, from 3.49 to 2.89 dB at 120 km/h and $L = 5$, and reduces the NOMA outage from 0.415 to 0.355. The PDR rises from 0.661 to 0.716 in that case and by about one percentage point at 60 km/h and $L = 2$; the gain grows with speed and CSI delay. SIC-ordering errors are rare (1–10% of the cluster-slots) because each cluster pairs users with very different gains, so most of the benefit comes from prediction-based rate selection rather than from reordering. A gap to perfect CSI remains, because the unpredictable part of shadowing cannot be removed by any predictor.

Figure 9 examines the NOMA model itself at $V = 100$ with the Head-UAV forced into NOMA, for cluster sizes $G = 2$, 3 and 4 and residual SIC factors from 0 to 0.2. With perfect SIC, NOMA reaches the PDR of Head-UAV OFDMA (0.870 vs. 0.871) with a 3.5% lower objective; with any residual interference it is worse, e.g., a PDR of 0.811 at $\varepsilon_{SIC} = 0.05$, and larger clusters do not help. A finer sweep at $V = 60$, 100 and 150 confirms this: NOMA

lowers the objective by 0.6–3.5% at $\varepsilon_{SIC} = 0$ but raises it from $\varepsilon_{SIC} = 0.005$ on, so the break-even point lies between 0 and 0.005. This agrees with the known result that OFDMA with bandwidth-proportional noise is close to the uplink sum capacity at high signal-to-noise ratio [29], while residual interference mainly hurts the weak user of each cluster. The access rule (2) therefore activates NOMA only for $\hat{\varepsilon}_{SIC} \leq \varepsilon_{th} = 0.0025$, and with the default $\varepsilon_{SIC} = 0.05$ the Head-UAV operates in OFDMA in all other experiments.

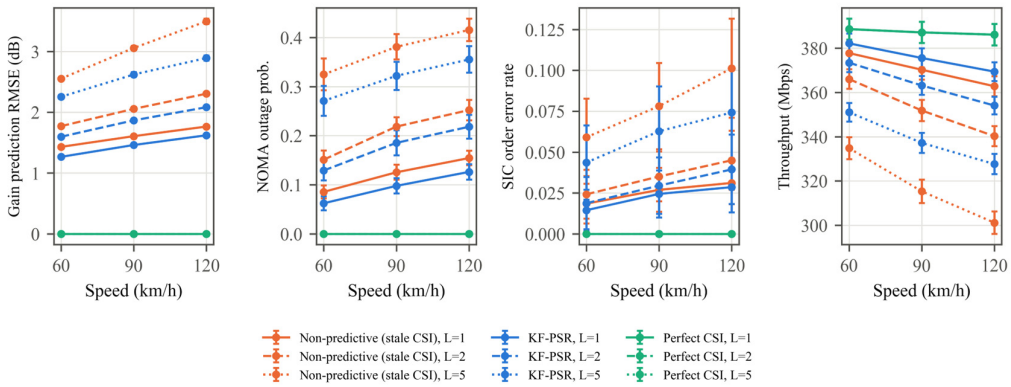


Figure 8. KF-PSR versus a non-predictive receiver and perfect CSI at $V = 100$ (Head-UAV in NOMA): gain-prediction RMSE, NOMA outage probability, SIC-order error rate and aggregate throughput versus vehicle speed for CSI delays $L = 1, 2$ and 5 slots. The legend below applies to all four subfigures. In the gain-prediction RMSE plot, the 95% confidence intervals are smaller than the markers (at most ± 0.04 dB). The Perfect CSI curves for $L = 1$, 2, and 5 overlap because perfect CSI is independent of the feedback delay L.

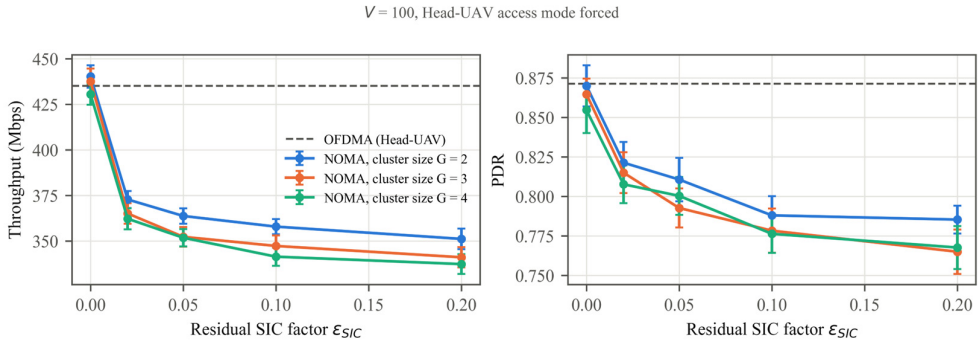


Figure 9. Head-UAV NOMA versus OFDMA at $V = 100$: aggregate throughput and PDR versus the residual SIC factor for cluster sizes $G = 2$, 3 and 4.

5.6. Ablation Study

Table 6 and Figure 10 remove one component at a time at $V = 60$ and 120. The iterative Bi-IE coupling is the main source of the optimizer's gain: without it (Phase 1 → Phase 2 → one re-allocation) the objective rises by 18% and the PDR falls from 0.754 to 0.596 at $V = 120$. Removing the MEC^{res} signal while keeping the alternation lowers the PDR to 0.729 ($p < 0.005$); the objective difference is not significant, so the signal adds a smaller gain in reliability at high load. KF-PSR raises the PDR at $V = 120$ from 0.731 to 0.754. The original SIC-unaware access rule, which activates NOMA at high load, lowers the PDR to 0.671, confirming the revised rule.

Table 6. Ablation of A-MASH (30 scenarios). F/F_{full} is the mean objective relative to that of the full framework (the paired per-scenario ratios are shown in Figure 10); * marks a PDR significantly different from the full framework (paired Wilcoxon test, $p < 0.05$).

Variant	V	F/F_{full}	PDR	P95 (s)	Energy (mJ)	Throughput (Mbps)
Full A-MASH	60	1.000	0.999	2.11	44.5	422
w/o Bi-IE (sequential)	60	1.011	0.996 *	2.29	47.5	405
AO-style coupling (no MEC^{res})	60	1.001	0.999	2.11	44.7	421
w/o battery-aware heuristic	60	1.000	0.999	2.11	44.4	422
w/o Overflow/Rescue (Phase 4)	60	1.000	0.999	2.11	44.5	422
w/o KF-PSR (stale CSI)	60	1.000	0.998	2.15	45.8	414
SIC-unaware access rule	60	1.034	0.996 *	2.30	48.6	369
Head fixed NOMA	60	1.044	0.995 *	2.37	49.9	353
Full A-MASH	120	1.000	0.754	3.32	60.2	439
w/o Bi-IE (sequential)	120	1.178	0.596 *	4.89	82.5	446
AO-style coupling (no MEC^{res})	120	1.013	0.729 *	3.52	63.8	432
w/o battery-aware heuristic	120	0.995	0.753	3.37	59.6	450
w/o Overflow/Rescue (Phase 4)	120	1.000	0.754	3.32	60.2	439
w/o KF-PSR (stale CSI)	120	1.000	0.731 *	3.40	62.2	427
SIC-unaware access rule	120	1.277	0.671 *	3.41	59.3	354
Head fixed NOMA	120	1.277	0.671 *	3.41	59.3	354

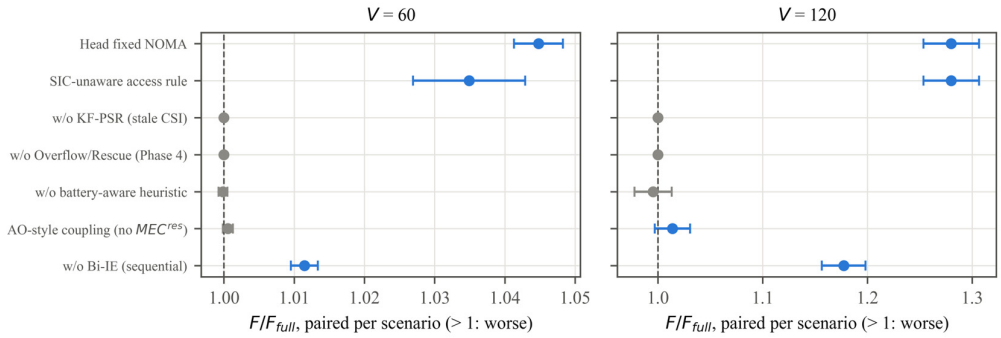


Figure 10. Planned objective value of each ablation variant relative to the full framework, paired per scenario, at $V = 60$ and 120 (mean and 95% confidence interval over 30 scenarios; values above 1 are worse; blue markers indicate that the 95% confidence interval excludes 1, and grey markers indicate that it includes 1). KF-PSR and the Overflow/Rescue controller act during execution and over missions, so their effects appear in the PDR of Table 6 and in Section 5.4 rather than here.

With the corrected model, the components that act within one decision no longer trade one metric against another: Bi-IE, the MEC^{res} signal and KF-PSR each improve the PDR and, at the same time, lower the latency and the vehicle energy at $V = 120$. The higher throughput without Bi-IE only reflects more admitted tasks (99% instead of 86%) that then miss their deadline. Similarly, the SIC-aware rule admits more tasks than the SIC-unaware rule (86% instead of 79%), which raises the mean latency and energy of the admitted tasks by 0.5% and 1.5% at $V = 120$, while it lowers both at $V = 60$.

Within a single epoch, the battery-aware heuristic and the Overflow/Rescue controller have no measurable effect: the Overflow triggers do not fire because the closed-form allocation uses the full CPU capacity of every UAV, and no UAV reaches its battery thresholds within one 30 s epoch. Their role is evaluated over full missions in Section 5.4. A-MASH selects the optional U2U relay for 59% of the tasks at $V = 10$ and for 7% at $V = 20$, and for no task from $V = 40$ on. A relay pays off only when the relay UAV has idle CPU capacity, which occurs only at low density; at higher density, every UAV uses its full CPU capacity,

so a relay would only add delay. Compared with A-MASH without the relay ($\xi = 0$), the relay lowers the mean latency at $V = 10$ by 5.9% (0.513 s vs. 0.545 s, $p < 0.001$); at $V = 20$, the difference is not significant. Overflow forwards occur only at $V = 10$ and 20 (0.7 and 0.6 tasks per decision) and in no A-MASH mission (Section 5.4).

5.7. Computational Budget, Convergence and Run Time

Figure 11 separates the quality of the optimizers from their budget. At $V = 100$ (Figure 11a), AO-BCD reaches its fixed point after about 1856 evaluations ($F = 0.896$) and gains nothing from a larger budget, whereas A-MASH keeps improving, from 0.920 with 1.5k evaluations to 0.858 with 6k and 0.849 with 12k. The standalone metaheuristics stay above $F = 2.19$ even with 12k evaluations. At $V = 150$ (Figure 11b), where A-MASH is behind AO-BCD and Greedy with the default budget, its objective improves from 2.147 (3k) to 1.895 (24k), against 1.802 for AO-BCD and 1.870 for Greedy, and its PDR from 0.607 to 0.645 (AO-BCD 0.672, Greedy 0.682). A larger budget therefore narrows the gap but does not close it against AO-BCD ($p < 0.01$ at every budget; against Greedy the difference is no longer significant at 24k evaluations): above the capacity bound most decisions concern admission, and the deterministic rejection rule of the two baselines is more effective than the ant-based search over admission decisions. Figure 11c shows the convergence at $V = 100$: because the first ant is built with the load-aware heuristic and the closed-form CPU split, the first solution of A-MASH is already better than the final solutions of the standalone metaheuristics in this scenario.

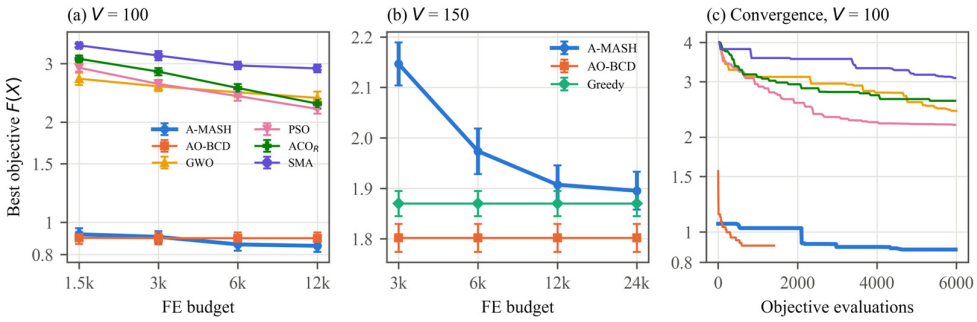


Figure 11. Best objective value versus the evaluation budget at (**a**) $V = 100$ and (**b**) $V = 150$, and (**c**) convergence at $V = 100$ with 6000 evaluations (one scenario). Subfigure (**c**) uses the legend of (**a**). AO-BCD stops at its fixed point (after about 1400 evaluations in (**c**)), and Greedy needs only a few evaluations, so their results do not change with the budget. Error bars are 95% confidence intervals over 30 scenarios.

Table 7 reports the run time per decision, measured in a single thread (Python/NumPy, 2.4 GHz laptop CPU, mean of 10 scenarios; 20 for $V = 100$, $K = 6$). A-MASH needs at most 2.2 s, about 7% of the 30 s epoch, and its run time grows sub-linearly with V and K because the budget is fixed. It is 4–6 times slower than a standalone metaheuristic with the same budget, owing to the sequential ant construction and the CPU re-splits. AO-BCD is faster for few UAVs but grows quickly with K, because each best-response move tries every UAV.

Table 7. Run time per decision (s), single thread, 6000 evaluations (AO-BCD stops at its fixed point).

Setting	A-MASH	AO-BCD	GWO	Greedy
$V = 20,\ K = 6$	1.01	0.12	0.19	<0.01
$V = 60,\ K = 6$	1.40	0.59	0.28	<0.01
$V = 100,\ K = 6$	1.78	0.51	0.36	0.01

Table 7. *Cont.*

Setting	A-MASH	AO-BCD	GWO	Greedy
$V = 150,\ K = 6$	2.19	0.71	0.48	0.02
$V = 100,\ K = 4$	1.60	0.33	0.36	0.01
$V = 100,\ K = 8$	1.93	1.47	0.38	0.01
$V = 100,\ K = 10$	2.10	1.98	0.36	0.01

5.8. Summary of the Sensitivity Analyses

Table 8 completes the sensitivity study with the objective weights, the deadline weight $\rho_{\mathbf{D}}$ and the battery-variance weight λ_{var}(A-MASH, at $V = 60$ and 120, 30 scenarios each). Shifting the weights towards latency, energy or battery, halving or doubling ρ_D, or setting λ_{var} to 0 or 2 changes the PDR by at most 0.6 percentage points and the latency and vehicle energy by at most 1.7%. The operating point is therefore robust to these weights, which need not be tuned finely: the soft-deadline term and the closed-form CPU split determine most decisions, and the battery terms differ little within one epoch.

Table 8. Sensitivity of A-MASH to the objective weights $(w_T, w_E, w_Q, w_B, w_S)$, the deadline weight $\rho_{\mathbf{D}}$ and the battery-variance weight λ_{var} (means over 30 scenarios).

Setting	V	PDR	Latency (s)	P95 (s)	Energy (mJ)	Admitted
Default (0.30, 0.25, 0.15, 0.20, 0.10), $\rho_D = 2$, $\lambda_{var} = 0.5$	60	0.999	1.56	2.11	44.5	100.0%
Latency emphasis (0.45, 0.20, 0.10, 0.15, 0.10)	60	0.998	1.56	2.12	45.0	100.0%
Energy emphasis (0.20, 0.45, 0.10, 0.15, 0.10)	60	0.998	1.56	2.10	44.4	100.0%
Battery emphasis (0.20, 0.20, 0.10, 0.40, 0.10)	60	0.999	1.56	2.12	44.6	100.0%
$\rho_D = 1$	60	0.999	1.56	2.12	44.7	100.0%
$\rho_D = 4$	60	0.999	1.56	2.10	44.5	100.0%
$\lambda_{var} = 0$	60	0.999	1.56	2.11	44.6	100.0%
$\lambda_{var} = 2$	60	0.999	1.56	2.10	44.5	100.0%
Default (0.30, 0.25, 0.15, 0.20, 0.10), $\rho_D = 2$, $\lambda_{var} = 0.5$	120	0.754	2.56	3.32	60.2	86.2%
Latency emphasis (0.45, 0.20, 0.10, 0.15, 0.10)	120	0.751	2.56	3.34	60.1	86.2%
Energy emphasis (0.20, 0.45, 0.10, 0.15, 0.10)	120	0.755	2.55	3.29	59.6	86.0%
Battery emphasis (0.20, 0.20, 0.10, 0.40, 0.10)	120	0.759	2.55	3.29	59.7	85.9%
$\rho_D = 1$	120	0.755	2.54	3.27	59.2	85.8%
$\rho_D = 4$	120	0.755	2.57	3.34	60.6	86.4%
$\lambda_{var} = 0$	120	0.751	2.56	3.33	60.2	86.2%
$\lambda_{var} = 2$	120	0.753	2.56	3.33	60.4	86.2%

The other sensitivity analyses of this section lead to four further findings. First, the gain of KF-PSR grows with speed and CSI delay: the PDR improves by 0.9 percentage points at 60 km/h and $L = 2$, and by 5.5 points at 120 km/h and $L = 5$ (Figure 8). Second, NOMA improves on OFDMA only with perfect SIC, and then only slightly (objective 0.6–3.5% lower, PDR within 0.3 percentage points); it is already worse at $\varepsilon_{SIC} = 0.005$, and larger clusters do not help (Figure 9). So, the access rule keeps the Head-UAV in OFDMA unless SIC is near-perfect ($\hat{\varepsilon}_{SIC} \leq \varepsilon_{th} = 0.0025$). Third, the per-vehicle service power is the most influential parameter: the first UAV withdrawal moves from 24.0 min at 0 W to 17.5 min at 2 W and 6.9 min at 15 W (Figure 7), so this power should be calibrated to the actual MEC payload. Fourth, A-MASH keeps improving with more evaluations and overtakes AO-BCD at $V = 100$ from 6k evaluations on, but does not reach it at $V = 150$ even with 24k evaluations (Figure 11). Above the capacity bound, even a four times larger budget therefore does not close the gap to AO-BCD, whose deterministic admission rule is more effective.

6. Conclusions

This paper proposed A-MASH, a hybrid optimization framework for a heterogeneous UAV-assisted IoV-MEC swarm in which a BS-connected Head-UAV cooperates with a FANET of Member-UAVs. A-MASH jointly decides association with admission control, CPU allocation, uplink power and U2U relaying by coupling a load- and battery-aware ACO with GWO through the Bi-IE mechanism, selects the Head-UAV access mode from load, battery and SIC quality, and supports decoding with the KF-PSR channel predictor. The evaluation used a slot-level simulator, an equal evaluation budget for all methods, 30 scenarios per density and paired significance tests. Under these conditions, A-MASH clearly outperforms standalone GWO, PSO, ACO_R and SMA from 40 vehicles on and performs on par with an alternating-optimization baseline up to 100 vehicles, while the iterative Bi-IE coupling is the main source of its gain.

Three limitations emerged. First, above the computing capacity of the swarm, simple model-based heuristics with deterministic admission are better, and a larger budget narrows but does not close the gap to the alternating-optimization baseline. Second, with order-dependent SIC, uplink NOMA improves on OFDMA only under near-perfect cancellation, and then only slightly, so the access rule keeps the Head-UAV in OFDMA for realistic SIC quality. Third, the swarm lifetime is dominated by hovering power: the battery-aware mechanisms postpone the first withdrawal by about one epoch, and the Overflow/Rescue controller trades a few points of delivery ratio for this protection. KF-PSR gives a consistent gain that grows with speed and CSI delay, and the results change by less than 2% when the objective weights are varied. Future work will seed the admission search with deterministic rules, add learning-based baselines, plan over several epochs to exploit battery heterogeneity, and include UAV trajectory control.

Funding: This research was supported by the Research Funds of Mokpo National University in 2025.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The simulation code, the configuration file and the raw per-run results used in this study are available on request from the corresponding author.

Acknowledgments: The author would like to thank the editor and the anonymous reviewers for their valuable comments and suggestions.

Conflicts of Interest: The author declares no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

A-MASH	Adaptive Mode-Aware Swarm Hybrid
ACO	Ant Colony Optimization
AO-BCD	Alternating Optimization/Block-Coordinate Descent
BLI	Battery Lifetime Index
Bi-IE	Bidirectional Information Exchange
CSI	Channel State Information
FANET	Flying Ad Hoc Network
FE	Objective-Function Evaluation
GWO	Grey Wolf Optimizer
IoV	Internet of Vehicles
KF-PSR	Kalman-Filter-based Prediction-assisted SIC ordering and Rate selection
KKT	Karush–Kuhn–Tucker
MEC	Mobile Edge Computing

NOMA	Non-Orthogonal Multiple Access
OFDMA	Orthogonal Frequency-Division Multiple Access
PDR	Packet (task) Delivery Ratio
PSO	Particle Swarm Optimization
SIC	Successive Interference Cancellation
SMA	Slime Mould Algorithm
U2U	UAV-to-UAV
UAV	Unmanned Aerial Vehicle

References

1. Peng, K.; Lin, Y.; Xu, X.; Yue, K.; Leung, V.C.M. Mobility-Aware Task Offloading in UAV-MEC-Assisted IoV: A Two-Stage Approach. *Ad Hoc Netw.* **2025**, *178*, 103956. [CrossRef]
2. Huda, S.M.A.; Moh, S. Survey on Computation Offloading in UAV-Enabled Mobile Edge Computing. *J. Netw. Comput. Appl.* **2022**, *201*, 103341. [CrossRef]
3. Peng, C.; Wang, Q.; Zhang, D. Efficient Dynamic Task Offloading and Resource Allocation in UAV-Assisted MEC for Large Sport Event. *Sci. Rep.* **2025**, *15*, 11828. [CrossRef]
4. Al-Hourani, A.; Kandeepan, S.; Lardner, S. Optimal LAP Altitude for Maximum Coverage. *IEEE Wirel. Commun. Lett.* **2014**, *3*, 569–572. [CrossRef]
5. Bekmezci, I.; Sahingoz, O.K.; Temel, Ş. Flying Ad-Hoc Networks (FANETs): A Survey. *Ad Hoc Netw.* **2013**, *11*, 1254–1270. [CrossRef]
6. Ding, Z.; Liu, Y.; Choi, J.; Sun, Q.; Elkashlan, M.; Chih-Lin, I.; Poor, H.V. Application of Non-Orthogonal Multiple Access in LTE and 5G Networks. *IEEE Commun. Mag.* **2017**, *55*, 185–191. [CrossRef]
7. Wu, Q.; Cui, M.; Zhang, G.; Zheng, B.; Chu, X.; Wu, Q. Energy-Efficient UAV-Enabled MEC Systems: NOMA, FDMA, or TDMA Offloading? *IEEE Trans. Veh. Technol.* **2026**, *75*, 8336–8352. [CrossRef]
8. Naeem, M.K.; Abozariba, R.; Asaduzzaman, M.; Patwary, M.N. Towards the Mobility Issues of 5G-NOMA Through User Dissociation and Re-association Control. In Proceedings of the 2020 IEEE 21st International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), Cork, Ireland, 31 August–3 September 2020; pp. 427–432. [CrossRef]
9. Naeem, M.K.; Abozariba, R.; Asaduzzaman, M.; Patwary, M.N. Mobility Support for MIMO-NOMA User Clustering in Next-Generation Wireless Networks. *IEEE Trans. Mob. Comput.* **2023**, *22*, 6011–6026. [CrossRef]
10. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
11. Socha, K.; Dorigo, M. Ant Colony Optimization for Continuous Domains. *Eur. J. Oper. Res.* **2008**, *185*, 1155–1173. [CrossRef]
12. Widians, J.A.; Wardoyo, R.; Hartati, S. A Hybrid Ant Colony and Grey Wolf Optimization Algorithm for Exploitation-Exploration Balance. *Emerg. Sci. J.* **2024**, *8*, 1642–1654. [CrossRef]
13. Zeng, Y.; Xu, J.; Zhang, R. Energy Minimization for Wireless Communication with Rotary-Wing UAV. *IEEE Trans. Wirel. Commun.* **2019**, *18*, 2329–2345. [CrossRef]
14. Aslam, N.; Wang, H.; Esmaiel, H.; Junejo, N.U.R.; Agamy, A. A Review of Wireless Charging Solutions for FANETs in IoT-Enabled Smart Environments. *Sensors* **2026**, *26*, 912. [CrossRef]
15. Barick, S.; Singhal, C. UAV-Assisted MEC Architecture for Collaborative Task Offloading in Urban IoT Environment. *IEEE Trans. Netw. Serv. Man.* **2025**, *22*, 732–743. [CrossRef]
16. Wang, L.; Shen, B.; Ma, L.; Zhang, Y.; Zhao, Y.; Guo, H.; Yu, Z.; Guo, B. Joint Task Offloading and Migration Optimization in UAV-Enabled Dynamic MEC Networks. *IEEE Trans. Serv. Comput.* **2025**, *18*, 2143–2157. [CrossRef]
17. Lakew, D.S.; Sa'ad, U.; Dao, N.N.; Na, W.; Cho, S. Routing in Flying Ad Hoc Networks: A Comprehensive Survey. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1071–1120. [CrossRef]
18. Dhall, R.; Dhongdi, S. Review of Protocol Stack Development of Flying Ad-Hoc Networks for Disaster Monitoring Applications. *Arch. Comput. Methods Eng.* **2023**, *30*, 37–68. [CrossRef]
19. Chen, J.; Ling, F.; Zhang, Y.; You, T.; Liu, Y.; Du, X. Coverage Path Planning of Heterogeneous Unmanned Aerial Vehicles Based on Ant Colony System. *Swarm Evol. Comput.* **2022**, *69*, 101005. [CrossRef]
20. Yang, S.; Wang, S.; Li, T.; Hu, T.; Xu, Z.; He, R.; Zhang, B. Hybrid Ant Colony-Based Inter-Cluster Routing Protocol for FANET. *Sci. Rep.* **2024**, *14*, 15632. [CrossRef]
21. Zheng, Z.; Huang, H.; Li, C.; Yu, Y.; Wang, X.; Cai, J.; Huang, X.; Hu, S. A Hybrid Search Behavior-Based Adaptive Grey Wolf Optimizer for Cooperative Path Planning for Multiple UAVs. *Sensors* **2025**, *25*, 7657. [CrossRef]
22. Aslam, Z.; Rahman, T.; Husnain, G.; Rehman, A.U.; Muniasamy, A.; Alkahtani, H.K. An Intelligent Bio-Inspired Multi-Objective and Scalable UAV-Assisted Clustering Algorithm in Flying Ad Hoc Networks. *Sci. Rep.* **2026**, *16*, 4870. [CrossRef]
23. Liu, J.; Hu, H.; Bai, X.; Li, G.; Zhang, X.; Zhou, H.; Li, H.; Liu, J. Multi-UAV-Assisted Task Offloading and Trajectory Optimization for Edge Computing via NOMA. *Sensors* **2025**, *25*, 4965. [CrossRef]

24. Qiu, H.; Gao, S.; Tu, G.; Zong, S. Position Information-Based NOMA for Downlink and Uplink Transmission in Mobile Scenarios. *IEEE Access* **2020**, *8*, 150808–150822. [CrossRef]
25. Jaiswal, R.K. Position-Based Routing Protocol Using Kalman Filter as a Prediction Module for Vehicular Ad Hoc Networks. *Comput. Electr. Eng.* **2020**, *83*, 106599. [CrossRef]
26. Li, S.; Hu, X.; Du, Y. Deep Reinforcement Learning and Game Theory for Computation Offloading in Dynamic Edge Computing Markets. *IEEE Access* **2021**, *9*, 121456–121466. [CrossRef]
27. Li, W.; Li, S.; Hao, J.; Wu, Q.; Wang, R. Efficient Task Offloading and Resource Allocation in HAPS-Assisted LEO Satellite Networks: A MAPPO With Exact Potential Game Approach. *IEEE Internet Things J.* **2026**, *13*, 10179–10195. [CrossRef]
28. ITU-R. *Propagation Data and Prediction Methods Required for the Design of Terrestrial Broadband Radio Access Systems Operating in a Frequency Range from 3 to 60 GHz*; Recommendation ITU-R P.1410; International Telecommunication Union: Geneva, Switzerland, 2023.
29. Tse, D.; Viswanath, P. *Fundamentals of Wireless Communication*; Cambridge University Press: Cambridge, UK, 2005.
30. Boyd, S.; Vandenberghe, L. *Convex Optimization*; Cambridge University Press: Cambridge, UK, 2004.
31. Gudmundson, M. Correlation Model for Shadow Fading in Mobile Radio Systems. *Electron. Lett.* **1991**, *27*, 2145–2146. [CrossRef]
32. Kalman, R.E. A New Approach to Linear Filtering and Prediction Problems. *J. Basic Eng.* **1960**, *82*, 35–45. [CrossRef]
33. Dorigo, M.; Maniezzo, V.; Colorni, A. Ant System: Optimization by a Colony of Cooperating Agents. *IEEE Trans. Syst. Man Cybern. B Cybern.* **1996**, *26*, 29–41. [CrossRef]
34. Kennedy, J.; Eberhart, R. Particle Swarm Optimization. In Proceedings of the IEEE International Conference on Neural Networks (ICNN'95), Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948. [CrossRef]
35. Li, S.; Chen, H.; Wang, M.; Heidari, A.A.; Mirjalili, S. Slime Mould Algorithm: A New Method for Stochastic Optimization. *Future Gener. Comput. Syst.* **2020**, *111*, 300–323. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.