

MAESTRO-X: Distributed Orchestration of Rotary-Wing UAV-Relay Swarms

Bharath Keshavamurthy*, Matthew A. Bliss†, and Nicolò Michelusi*

Abstract

This work details a scalable framework to orchestrate a swarm of rotary-wing UAVs serving as cellular relays to facilitate beyond line-of-sight connectivity and traffic offloading for ground users. First, a Multiscale Adaptive Energy-conscious Scheduling and TRajecotry Optimization (MAESTRO) framework is developed for a single UAV. Aiming to minimize the time-averaged latency to serve user requests, subject to an average UAV power constraint, it is shown that the optimization problem can be cast as a semi-Markov decision process, and exhibits a multiscale structure: outer actions on radial wait velocities and terminal service positions minimize the long-term delay-power trade-off, optimized via value iteration; given these outer actions, inner actions on angular wait velocities and service trajectories minimize a short-term delay-energy cost; finally, rate adaptation is embedded along the trajectory to leverage air-to-ground channel propagation conditions. A novel hierarchical competitive swarm optimization scheme is developed in the inner optimization, to devise high-resolution trajectories via iterative pair-wise updates. Next, MAESTRO is eXtended to UAV swarms (MAESTRO-X) via scalable policy replication, enabled by a decentralized command-and-control network augmented with: (1) *spread maximization* to proactively position UAVs to serve future requests; (2) *consensus-driven conflict resolution* to orchestrate scheduling decisions based on delay-energy costs including queuing dynamics; (3) *adaptive frequency reuse* to improve spectrum utilization across the network; and (4) a *piggybacking mechanism* allowing UAVs to serve multiple ground users simultaneously. Numerical evaluations show that, for user requests of 10 Mbits, generated according to a Poisson arrival process with rate 0.2 req/min/UAV, single-agent MAESTRO offers $3.8\times$ faster service than a high-altitude platform and 29% faster than a static UAV deployment; moreover, for a swarm of 3 UAV-relays, MAESTRO-X delivers data payloads $4.7\times$ faster than a successive convex approximation scheme; and remarkably, a single UAV optimized via MAESTRO outclasses 3 UAVs optimized via a deep-Q network by 38%.

Index Terms

UAV-Relays, Trajectory optimization, SMDPs, Hierarchical CSO

A preliminary version of this work was presented at Asilomar 2022 [1]. Source code is available on GitHub [2].

Part of this work has been supported by NSF under grants CNS-1642982 and CNS-2129015.

*Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ.

†Electrical and Computer Engineering, Purdue University, West Lafayette, IN.

I. INTRODUCTION

Enterprises across various industrial sectors have stepped-up the adoption of Unmanned Aerial Vehicles (UAVs) to gather data, survey infrastructure, monitor operations, and automate logistics [3], [4]. UAVs can also be leveraged to enhance troop deployments in military scenarios [5], aid emergency response during a natural disaster [6], and facilitate data harvesting in precision agriculture [7]. Inevitably, this has fostered varied academic research and industrial R&D on UAV-augmented beyond line-of-sight connectivity and traffic offloading in cellular networks, whose coverage can be enhanced by the mobility and maneuverability of UAVs [8], [9].

Yet, the pervasive potential of UAV-assisted wireless networks presents a plethora of challenges in real-world deployments [9]: limited on-board energy of aerial platforms, Quality-of-Service (QoS) requirements, air-to-ground channels, and computational feasibility challenges of UAV trajectory design. Several works have tackled some of these challenges by employing tools from optimization and artificial intelligence—however, numerous problems remain unsolved: failure to capture uncertain system dynamics vis-à-vis random traffic arrivals [10]–[14]; restrictions on UAV path and velocity characteristics [11], [15]; inefficient centralized swarm deployments [16]–[18]; computationally expensive joint multi-agent formulations offering limited scalability [19]–[22]; and failure to account for link layer effects on the QoS of the network [23], [24].

In this paper, considering these drawbacks in the state-of-the-art, we study the decentralized orchestration of multiple power-constrained rotary-wing UAVs supplementing a terrestrial base station by relaying data traffic dynamically generated by ground users. Incorporating waiting state optimization, computationally feasible trajectory design, throughput-maximizing rate adaptation to Air-to-Ground (A2G) propagation conditions, queue management, frequency reuse to enhance spectrum utilization, multi-user service, and multi-UAV consensus-driven scheduling, we develop a scalable framework to efficiently automate the operations of distributed UAV-relay deployments.

Ergo, specializing to single UAV-relay settings, we first propose MAESTRO, a Multiscale Adaptive Energy-conscious Scheduling and TRajecotry Optimization framework to control the idle and service phase operations of the UAV. Seeking to minimize the average communication delay subject to an average UAV mobility power constraint, we show that the problem can be cast as a Semi-Markov Decision Process (SMDP) with a multiscale structure: outer decisions on radial velocities and terminal service positions influence the long-term delay-power cost; consequently, given these outer actions, inner actions on angular wait velocities and service

Paper	Adaptive control	Channel model	Frequency reuse	Multiuser service	UAV Motion Mobility	UAV Velocity	UAV deployment	Multi-UAV scheduling	Overall formulation	Link Layer Schedule	Queue
MAESTRO-X	Yes	A2G	Yes	Yes	Dynamic	Variable	Distributed	Decoupled	Model-based	Yes	Yes
[10]	No	FSPL	No	No	Dynamic	Variable	Single	-	Model-based	Yes	No
[16]	No	A2G	Yes	Yes	Dynamic	Variable	Centralized	Joint	Model-based	Yes	No
[19]	No	A2G	No	Yes	Restricted	Fixed	Distributed	Joint	Model-free	No	No
[11]	No	FSPL	No	No	Dynamic	Fixed	Single	-	Model-based	Yes	No
[12]	No	FSPL	No	No	Dynamic	Variable	Single	-	Model-based	Yes	No
[20]	No	FSPL	No	Yes	Restricted	Fixed	Distributed	Joint	Model-free	Yes	No
[13]	No	A2G	No	No	Static	-	Single	-	Model-based	No	No
[23]	No	FSPL	No	No	Static	-	Distributed	Joint	Model-based	Yes	No
[24]	Yes	FSPL	No	No	Static	-	Distributed	Joint	Model-based	No	No
[17]	No	FSPL	No	No	Dynamic	Fixed	Centralized	Joint	Model-based	Yes	No
[18]	No	A2G	No	No	Static	-	Centralized	Joint	Model-based	No	No
[27]	No	A2G	No	No	Restricted	Fixed	Distributed	Decoupled	Model-free	No	No
[21]	Yes	FSPL	No	No	Static	-	Distributed	Joint	Model-free	No	Yes
[22]	Yes	A2G	No	No	Static	-	Distributed	Joint	Model-free	No	No
[14]	No	A2G	No	No	Dynamic	Variable	Single	-	Model-based	Yes	No
[28]	Yes	FSPL	No	No	Dynamic	Variable	Single	-	Model-free	No	No

TABLE I: A comparison of the features of our framework with those of relevant schemes in the literature.

trajectories minimize a short-term delay-energy cost. We develop a value iteration algorithm [25] exploiting this multiscale structure to optimize outer actions, and a hierarchical variant of Competitive Swarm Optimization (CSO) [26], decoupled from value iteration, to optimize high-resolution trajectories embedding a novel throughput maximizing rate adaptation scheme for A2G channels. Next, we extend MAESTRO to a swarm of UAV-relays (MAESTRO-X) via a scalable replication strategy, enabled by a decentralized command-and-control network and augmented with: spread maximization to proactively position the UAVs to serve future service requests; consensus-driven conflict resolution to orchestrate ground user scheduling decisions based on delay-energy costs, including queuing dynamics; frequency reuse to enhance spectrum utilization; and piggybacking to enable each UAV to serve multiple users simultaneously.

Related Work: Table I summarizes our approach (MAESTRO-X) and contrasts it with relevant works in the state-of-the-art. First, we observe non-adaptive schemes, e.g., [10], [17], [18] designed for applications where ground users possess local storage or aggregation capabilities allowing for deterministic traffic; however, practical deployments involve dynamically generated requests and randomly located ground users. Accommodating these uncertainties calls for the design of adaptive UAV orchestration frameworks. Yet, existing works do so only for single UAV-relay deployments [28] or consider static placement of UAVs (i.e., no trajectory design) [21], [22], [24]. In contrast, we design adaptive trajectory and scheduling strategies for distributed multi-UAV swarms, that accommodate dynamic and uncertain traffic generated by ground users.

Next, works employing Free Space Pathloss (FSPL) channel models, e.g., [10]–[12], [20], fail to account for the A2G channel characteristics in UAV-assisted wireless networks. Existing works that model A2G channels fail to leverage small- and large-scale A2G conditions via

rate adaptation. A notable exception is [14], which differs from our rate adaptation scheme in two ways: 1) we select the rate to maximize throughput (vs. [14], which aims to satisfy an outage constraint), and 2) we use a probabilistic line-of-sight (LoS) and Non-LoS (NLoS) model. Furthermore, most works surveyed neither consider spectrum reuse (with the exception of [16]) nor permit simultaneous multi-user service (with the exception of [16], [19], [20])—however, the works that do incorporate these crucial features [16], [19], [20] fail to consider adaptation to dynamically generated requests from randomly located users, as done in our work.

A common approach for trajectory design is Successive Convex Approximation (SCA) [10], [14]. SCA typically relies on the FSPL channel model to devise convex relaxations of the objective and constraints. Exceptions include [14] and [16], which apply SCA approaches under A2G channels. In [14], a logistic approximation of the achievable rate is used under outage constraints; in [16], only large-scale fading is considered. However, when coupling trajectory design with our throughput-maximizing rate adaptation scheme, closed-form rate expressions with first-order convex approximations are impractical. To tackle this challenge, we propose a CSO [26] approach for UAV trajectory design. **Unlike SCA, CSO does not rely on the problem structure of FSPL models to work effectively, and can thus accommodate realistic A2G propagation conditions.** Particle Swarm Optimization (PSO) [29], a swarm-based optimization method in which particle updates are driven by the global and individual best positions, has been used to optimize static UAV placement [30], [31], or restricted UAV trajectories (e.g., moving along a circle [15], or with fixed speed [11]). Removing these restrictions calls for the more efficient update strategy of CSO, which exhibits superior performance on several benchmarks [26]: it involves pair-wise particle competitions, wherein winners advance to the next iteration and the losers learn from the winners. **Moreover, we scale CSO to higher-dimensional trajectory design by embedding it within a Hierarchical wrapper (HCSO), which iteratively optimizes trajectories of increasing resolution, without imposing unreasonable restrictions on UAV mobility.**

Next, shifting our attention to swarm orchestration frameworks, several approaches consider centralized multi-UAV deployments [16]–[18] in which an aggregation center coordinates the UAV-relaying operations; or either joint multi-relay solutions [16], [23], [24] or model-free formulations constituting combined state and action spaces [19]–[22]. **An exception is [27], which considers a model-free setup with decentralized UAV deployments and decoupled scheduling.** But, [27] does not consider adaptation to randomly-generated data traffic, as we do in our work; rather, a sense-and-send protocol is devised, wherein tasks are always ready to be sensed.

Centralized swarm deployments often need additional capital and operational expenditure, and joint multi-UAV designs lead to large solution spaces resulting in prohibitive convergence times. Mindful of such considerations, we present an orchestration framework suitable for distributed UAV deployments by replicating our single-agent policy across the swarm and augmenting it with spread maximization and consensus-driven link-layer prescient conflict resolution over a command-and-control network. This eliminates the need for a centralized aggregation center, mitigates the computational overhead encountered by joint multi-relay models, and facilitates the seamless incorporation of queuing dynamics into scheduling decisions. Also, as shown in our numerical evaluations, our framework can be scaled to networks with ≥ 10 UAVs, while state-of-the-art approaches [10], [16], [19] become prohibitively expensive for networks with 5 UAVs. Additionally, although model-free control schemes [19]–[22], [27], [28] consider unknown system dynamics when solving for the optimal trajectory and/or scheduling solution, they fail to efficiently exploit the problem structure, resulting in large policy convergence times. In contrast, we use a model-based approach, by casting the problem as an SMDP, which captures the temporal irregularities seen in the state transitions of UAV-augmented wireless networks.

Contributions: We develop a novel framework for the scalable orchestration of UAV-relay swarms. To the best of our knowledge, no other work simultaneously incorporates the practical features of 1) dynamic traffic from randomly located ground users; 2) efficient exploitation of A2G channel conditions via a throughput-maximizing rate adaptation scheme; 3) easy scalability to large UAV swarms via policy replication, coupled with multi-agent coordination mechanisms over a distributed command-and-control network; and 4) waiting state optimization to position idle UAVs for potential new requests. In a nutshell, the contributions of this paper are:

- **MAESTRO**: For a single UAV, we construct an adaptive scheduling and trajectory design framework to minimize the communication latencies in serving dynamic transmission requests generated by randomly located ground users, subject to an average UAV power constraint. We show that the problem can be solved as a *Semi-Markov Decision Process* (SMDP). A multiscale decomposition facilitates efficient computation of rate adaptation, scheduling and trajectory solutions, and energy-conscious orchestration of the UAV during idle periods.
- **HCSO**: To enable computationally tractable design of high-resolution UAV trajectories under A2G propagation conditions, we propose *Hierarchical CSO* (HCSO), a variant of CSO wherein iterative pair-wise cost comparisons devise trajectories of increasingly higher resolution.
- **MAESTRO-X**: Coupled with decentralized command-and-control operations over a distributed

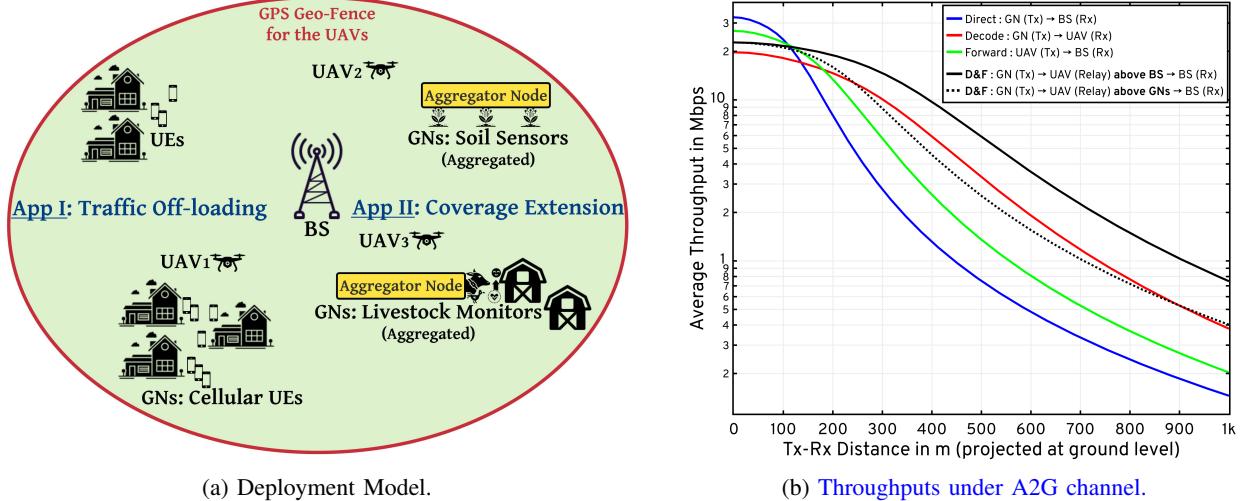


Fig. 1: (a) A terrestrial BS aided by UAVs serving as relays for a diverse set of GNs: traffic offloading for cellular UEs, and coverage extensions for livestock monitors and soil sensors; (b) rate-adapted throughputs (see Table II for the numerical parameters) along the GN→BS link (*direct*), GN→UAV link (*decode*), UAV→BS link (*forward*), and GN→UAV→BS link (*decode-and-forward*, with the UAV relay stationed above the BS or the GN).

mesh network, we augment the single-UAV trained policy with multi-UAV mechanisms to orchestrate waiting phase operations (*spread maximization*), coordinate scheduling decisions incorporating queuing dynamics (*consensus-driven conflict resolution*), enable simultaneous multi-user service (*piggybacking*), and enhance spectrum utilization (*frequency reuse*).

The rest of the paper is organized as follows: Sec. II introduces the system model; Sec. III elucidates the design of MAESTRO; Sec. IV describes the main algorithms; Sec. V details policy replication and multi-UAV mechanisms to manage distributed swarms (MAESTRO-X); Sec. VI chronicles our numerical evaluations; and finally, Sec. VII lists our concluding remarks.

II. SYSTEM MODEL

Consider the deployment scenario depicted in Fig. 1a: a swarm of N_U rotary-wing Unmanned Aerial Vehicles (UAVs) operate as cellular relays to supplement a terrestrial Base Station (BS) by relaying data traffic dynamically generated by Ground Nodes (GNs). The BS is located at the center of the circular cell (of radius a), at height H_B . The UAVs operate at a fixed height H_U . The GNs are distributed uniformly at random throughout the cell, with density λ_G [GNs/unit area]. Multi-user communication is enabled via OFDMA over a spectrum of bandwidth W , discretized into N_C orthogonal data channels (possibly, obtained by grouping multiple subcarriers together), each with bandwidth $B \triangleq \frac{W}{N_C}$. We assume the system operates in the uplink, i.e., traffic requests generated by the GNs are transmitted to the BS, either directly or by using one UAV as a relay. It can be extended to both uplink/downlink via a state variable differentiating between the two.

Communication Model: Each GN generates uplink transmission requests of L bits, according to a Poisson process with rate $\lambda_{R|G}$ [requests/GN/unit time]. Coupled with the random deployment of GNs, uplink requests arrive in time according to a Poisson process with rate $\Lambda \triangleq \lambda_G \cdot \lambda_{R|G} \pi a^2$ [requests/unit time] over the circular cell. Since a new request is uniformly distributed in the cell area, the position (r, θ) of the source GN—expressed in polar coordinates with respect to the BS—has angular coordinate θ uniform in $[0, 2\pi]$, and radial coordinate with probability density function given by $f_R(r) = \frac{2r}{a^2} \mathbb{I}(r \leq a)$, where $\mathbb{I}(\cdot)$ is the indicator function.

A fully-connected mesh network overlaying the BS and UAVs enables command-and-control using the band-edges of the allocated spectrum as control channels. Since control packets constitute short frames relative to the large GN-generated data payloads (communicated over data channels), the control operation latencies are neglected. To request uplink transmission to the BS, a GN sends a service request with its location; the BS broadcasts this *need-for-service* to the UAV swarm. Next, a consensus-driven conflict resolution process occurs among the BS and all UAVs (Sec. V), based on assessed delay-energy costs for this request, culminating in a scheduling decision. If direct-BS transmission is chosen, the BS chooses an available data channel, or queues the request until one becomes available (see Sec. V). The BS then instructs the GN to begin direct transmission over the data channel. Otherwise, if UAV relay i is selected, the new GN request is served via a *Decode-and-Forward* (D&F) strategy on an available data channel (or queued until one becomes available), as detailed in Sec. V. While executing the D&F protocol, the UAV moves along a pre-designed energy-conscious trajectory, i.e., a sequence of way-points and velocities (see Sec. IV). In Sec. V, we also discuss a *frequency reuse* mechanism to improve spectrum utilization efficiency, and a *piggybacking* mechanism allowing the scheduled UAV to serve multiple requests simultaneously. As evident from this communication model, the $\text{GN} \rightarrow \text{BS}$, $\text{GN} \rightarrow \text{UAV}$, and $\text{UAV} \rightarrow \text{BS}$ links must be characterized, as detailed next.

A2G Channel Model: For a generic link, we denote the flat-fading channel coefficient as $h \triangleq \sqrt{\beta}g$, where β captures the large-scale channel variations, and g with $\mathbb{E}[|g|^2] = 1$ denotes the small-scale fading component. We model the large-scale component as $\beta = \beta_{\text{LoS}}(d) \triangleq \beta_0 d^{-\alpha}$ for LoS and $\beta = \beta_{\text{NLoS}}(d) \triangleq \kappa \beta_0 d^{-\tilde{\alpha}}$ for NLoS links, where β_0 is the pathloss at a reference distance of 1 m, $2 \leq \alpha \leq \tilde{\alpha}$ are the LoS and NLoS pathloss exponents, $\kappa \in (0, 1]$ captures the additional NLoS attenuation, and d denotes the Tx-Rx Euclidean distance [10]. Following [32], we use a probabilistic LoS model, with LoS probability $P_{\text{LoS}}(\varphi) = [1 + z_1 \exp\{-z_2(\varphi - z_1)\}]^{-1}$, where $\varphi \in (0^\circ, 90^\circ]$ is the Tx-Rx elevation angle, and z_1, z_2 are parameters specific to the

propagation environment (e.g., urban, suburban, rural) [32]. The distribution of the small-scale fading component g also depends on the LoS or NLoS link state [33]: for LoS links, as in [14], we model g as Rician fading with a φ -dependent K -factor $K(\varphi)=k_1 \exp\{k_2 \varphi\}$, where k_1, k_2 are specific to the propagation environment; for NLoS links, we model g as Rayleigh fading (Rician with $K=0$) [33]. Given h , the link capacity is $C(h)=B \cdot \log_2 \left(1+\frac{|h|^2 P}{N_0 B}\right)$, where P is the transmission power, N_0 is the noise power spectral density at the receiver, and B is the channel bandwidth. We assume that other sources of signal degradation, such as the Doppler effect, are well-compensated at the receiver (for example, see the approaches in [34]).

Since the large-scale fading components typically vary slowly relative to the acquisition rate of Channel State Information (CSI), we assume that the current large-scale parameters (β, K) are known at the transmitter's side throughout the communication process, using CSI feedback over the control channel. Conversely, small-scale fading conditions vary at a much faster timescale and cannot be tracked at the transmitter. Hence, given (β, K) and a transmission rate Υ [bits/second], we define the outage probability as $P_{\text{out}}(\Upsilon, \beta, K) \triangleq \mathbb{P}(C(\sqrt{\beta}g) < \Upsilon) | \beta, K = \mathbb{P}(|g|^2 < u(\Upsilon, \beta))$, where $u(\Upsilon, \beta) \triangleq \frac{N_0 B}{\beta P} (2^{\frac{\Upsilon}{B}} - 1)$. The expected throughput is then $R(\Upsilon, \beta, K) = \Upsilon \cdot (1 - P_{\text{out}}(\Upsilon, \beta, K))$, assuming that the small-scale fading is averaged out across time. The rate Υ is then selected to maximize the expected throughput (as opposed to the approach in [14], which imposes an outage probability constraint) as $\Upsilon^*(\beta, K) \triangleq \arg \max_{\Upsilon \geq 0} R(\Upsilon, \beta, K)$, solved in Proposition 1.

Proposition 1. Given the large-scale parameters (β, K) and $\gamma \triangleq \frac{N_0 B}{\beta P}$, the optimal throughput-maximizing rate is $\Upsilon^*(\beta, K) = B \log_2 \left(1 + \frac{Z^*}{2}\right)$, where Z^* is the unique solution in $(0, \infty)$ of

$$h'(Z) \triangleq \frac{1}{(2+Z) \ln \left(1 + \frac{Z}{2}\right)} - \frac{\gamma(K+1)e^{-K}}{2} \frac{\exp\{-\gamma(K+1)\frac{Z}{2}\} I_0(\sqrt{2\gamma K(K+1)Z})}{Q_1(\sqrt{2K}, \sqrt{\gamma(K+1)Z})} = 0, \quad (1)$$

where $I_0(x)$ is the modified Bessel function of first kind of order 0, $Q_1(\cdot, \cdot)$ is the standard Marcum Q -function [14]. Z^* is solvable via the bisection method. The expected throughput is

$$R^*(\beta, K) \triangleq \max_{\Upsilon \geq 0} R(\Upsilon, \beta, K) = \Upsilon^*(\beta, K) \cdot Q_1(\sqrt{2K}, \sqrt{2(K+1)u(\Upsilon^*(\beta, K), \beta)}). \quad (2)$$

Proof. See Appendix A. ■

When $K=0$ (Rayleigh fading for NLoS), Q_1 specializes to $Q_1(0, \sqrt{2u(\Upsilon, \beta)}) = \exp\{-u(\Upsilon, \beta)\}$, while the condition $h'(Z)=0$ becomes $(1 + \frac{Z}{2}) \ln(1 + \frac{Z}{2}) = \frac{1}{\gamma}$. Finally, with the LoS and NLoS conditions averaged out in the temporal and spatial dimensions, the average link throughput is

$$\bar{R}(d, \varphi) \triangleq P_{\text{LoS}}(\varphi) \cdot R^*(\beta_{\text{LoS}}(d), K(\varphi)) + (1 - P_{\text{LoS}}(\varphi)) \cdot R^*(\beta_{\text{NLoS}}(d), 0). \quad (3)$$

This expression is then specialized to the three distinct communication links by expressing the transmission powers, the environment-specific parameters (z_1, z_2, k_1, k_2), the large-scale parameters (β, K), and the LoS/NLoS probabilities based on the spatial configuration, i.e., d and φ . For the GN→BS link, we let $\bar{R}_{GB}(r)$ be the throughput with the GN in position (r, θ) , computed by setting the GN-BS distance as $d = \sqrt{H_B^2 + r^2}$ and the elevation angle as $\varphi = \sin^{-1}(\frac{H_B}{d})$ in (3). Similarly, for the GN→UAV link, we let $\bar{R}_{GU}(r_{GU})$ be the throughput when the GN-UAV distance (projected onto the $x-y$ plane) is r_{GU} , computed by setting the GN-UAV Euclidean distance as $d = \sqrt{r_{GU}^2 + H_U^2}$ and the elevation angle as $\varphi = \sin^{-1}(\frac{H_U}{d})$ in (3). Finally, for the UAV→BS link, we let $\bar{R}_{UB}(r_{UB})$ be the throughput when the $x-y$ projected UAV-BS distance is r_{UB} , computed by setting the GN-UAV Euclidean distance as $d = \sqrt{r_{UB}^2 + (H_U - H_B)^2}$ and the elevation angle as $\varphi = \sin^{-1}(\frac{H_U - H_B}{d})$ in (3). As shown in Figs. 1b, the poor QoS experienced by GNs farther away from the BS, caused by deterioration in LoS probabilities with distance, motivates the need for UAV-relays to improve coverage throughout the cell.

UAV Mobility Power Model: For a rotary-wing UAV, since its communication power needs (≈ 10 W) are dwarfed by its mobility power requirements (≈ 1000 W), we model the on-board energy expenditure as a function of the horizontal flying velocity V [10], i.e.,

$$P_{\text{mob}}(V) = P_1 \left(1 + \frac{3V^2}{U_{\text{tip}}^2} \right) + P_2 \left(\sqrt{1 + \frac{V^4}{4v_0^4}} - \frac{V^2}{2v_0^2} \right)^{0.5} + P_3 V^3, \quad 0 \leq V \leq V_{\max}, \quad (4)$$

where P_i are the scaling constants, U_{tip} is the rotor blade tip velocity, v_0 is the mean rotor induced velocity while hovering, and V_{\max} is the maximum UAV flying speed [10]. We let $P_{\max} \triangleq \max_{0 \leq V \leq V_{\max}} P_{\text{mob}}(V)$ and $P_{\min} \triangleq \min_{0 \leq V \leq V_{\max}} P_{\text{mob}}(V)$ be the maximum and minimum power consumption of the UAV, respectively. From [10], hovering requires $P_{\text{mob}}(0) = 1371$ W, while flying at 22 m/s only consumes $P_{\min} = 936$ W. This suggests that the mobility of the UAVs can be exploited to reduce power consumption, while simultaneously improving coverage across the cell. Our goal is to define an energy-conscious adaptive service scheduling and trajectory optimization scheme to minimize the time-averaged communication delay experienced by GNs in the cell, subject to an average per-UAV mobility power constraint, studied next.

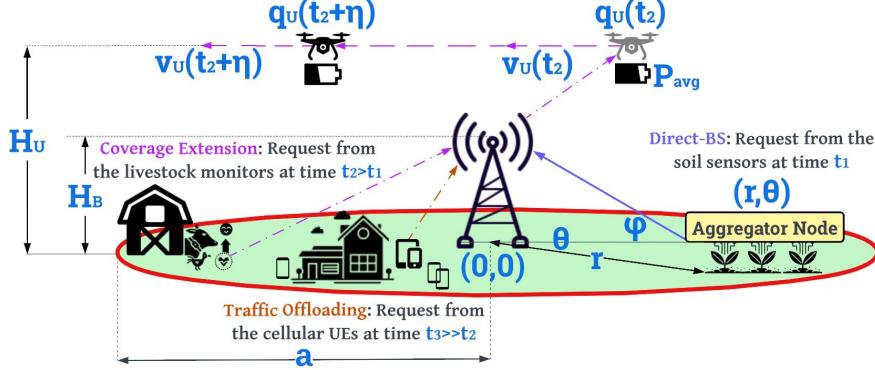


Fig. 2: The single-agent specialization of our generalized deployment depicted in Fig. 1a.

III. MAESTRO: A SEMI-MARKOV DECISION PROCESS FORMULATION

We now specialize the system model to a single UAV relay (illustrated in Fig. 2) via an SMDP formulation. The effective traffic rate experienced by a single UAV is $\Lambda' \triangleq \frac{\Lambda}{N_U}$ [requests/unit time/UAV], assumed in this section in place of the overall rate Λ . Let $\mathbf{q}_U(t) = (r_U(t), \theta_U(t))$ be the polar coordinate of the UAV at time t , projected onto the $x-y$ plane, where $r_U(t) \in \mathbb{R}_+$ and $\theta_U(t) \in [0, 2\pi)$ denote the UAV's radius and angle with respect to the BS. The system operates with the following phases. In the *waiting phase*, no GN requests are being served by the UAV, which moves according to a *waiting policy*. When a new GN request originates in position (r, θ) , the system transitions to the *request scheduling phase*, where it is determined whether the GN should transmit its data payload directly to the BS, or relay it through the UAV. In case of direct transmission, the system immediately re-enters the waiting phase, as the UAV remains free to serve other requests; else, the system enters the *UAV relay phase*, in which the data payload is relayed through the UAV using the D&F protocol; upon completion, the system re-enters the waiting phase. In this section, we conservatively assume that: 1) when the UAV is serving a request, it is unable to serve other incoming requests, which are thus directly served by the BS; and 2) data channels are always available at the BS to serve incoming requests. We defer to Sec. V for the description of a *piggybacking* mechanism to simultaneously serve multiple transmission requests, and of a queuing mechanism when data channels are unavailable.

Communication Delay and UAV Energy Consumption: Here, we formulate the average communication delay and UAV energy consumption under a given policy μ that defines the request scheduling, communication strategy, and UAV trajectory (formally defined later). We define a decision interval as the time duration spanning the start of a waiting phase, the subsequent request scheduling phase when a GN request is received, until the system re-enters the waiting phase after scheduling a direct transmission to the BS, or following the UAV relay phase.

Consider the u th such decision interval of duration Δ_u , split into the time $\Delta_u^{(w)}$ to wait for a new request, and the time $\Delta_u^{(s)}$ to serve it, either through the BS (scheduling decision $\xi_u=0$) or through the UAV ($\xi_u=1$). Then, $\Delta_u=\Delta_u^{(w)}+\xi_u\Delta_u^{(s)}$, since the UAV enters the waiting phase immediately (and the decision interval terminates) in case of direct-BS transmission. Let $N_u \geq 0$ be the number of additional requests received during the UAV relay phase of the u th decision period: since these are served directly by the BS, we denote their delays as $\Delta_{u,i}^{(bs)}$, $i=\{1, 2, \dots, N_u\}$. Let E_u be the UAV mobility energy expended during the u th decision interval, and let M_t be the total number of decision intervals completed up to time t . We define the expected long-term average communication delay per request (\bar{D}_μ) and average UAV power (\bar{P}_μ), under μ , as

$$\bar{D}_\mu \triangleq \lim_{t \rightarrow \infty} \mathbb{E}_\mu \left[\frac{\frac{1}{M_t} \sum_{u=1}^{M_t} (\Delta_u^{(s)} + \xi_u \sum_{i=1}^{N_u} \Delta_{u,i}^{(bs)})}{\frac{1}{M_t} \sum_{u=1}^{M_t} (1 + \xi_u N_u)} \right], \quad \bar{P}_\mu \triangleq \lim_{t \rightarrow \infty} \mathbb{E}_\mu \left[\frac{\frac{1}{M_t} \sum_{u=1}^{M_t} E_u}{\frac{1}{M_t} \sum_{u=1}^{M_t} \Delta_u} \right]. \quad (5)$$

Note that \bar{D}_μ in (5) captures the delays of all requests, i.e., those relayed through the UAV ($\xi_u=1$), those transmitted directly to the BS ($\xi_u=0$), as well as the N_u additional requests served directly by the BS during the UAV relay phase. Thus, the objective is to solve

$$\bar{D}^* = \min_{\mu} \bar{D}_\mu, \text{ s.t. } \bar{P}_\mu \leq P_{\text{avg}}, \quad (6)$$

where $P_{\text{avg}} \in (P_{\text{min}}, P_{\text{max}})$ is the average power constraint, and the optimal policy is denoted as μ^* . To simplify, let $\bar{\mathbb{E}}_\mu[C_u] \triangleq \lim_{t \rightarrow \infty} \mathbb{E}_\mu[\frac{1}{M_t} \sum_{u=1}^{M_t} C_u]$ be a shorthand notation for the long-term average cost C_u per decision interval. Let $\bar{E}_\mu \triangleq \bar{\mathbb{E}}_\mu[E_u]$ be the average UAV energy expenditure, $\bar{T}_\mu \triangleq \bar{\mathbb{E}}_\mu[\Delta_u]$ be the average interval duration, $\bar{N}_\mu \triangleq \bar{\mathbb{E}}_\mu[1 + \xi_u N_u]$ be the average number of requests served, $\bar{W}_\mu^{(s)} \triangleq \bar{\mathbb{E}}_\mu[\Delta_u^{(s)}]$ be the average delay of requests for which a scheduling decision is made, $\bar{W}_\mu^{(bs)} \triangleq \bar{\mathbb{E}}_\mu[\xi_u \sum_{i=1}^{N_u} \Delta_{u,i}^{(bs)}]$ be the average delay of requests served directly by the BS during the UAV relay phase, per decision interval. Using Little's Law [35], we can then express $\bar{P}_\mu = \frac{\bar{E}_\mu}{\bar{T}_\mu}$ and $\bar{D}_\mu = \frac{\bar{W}_\mu^{(s)} + \bar{W}_\mu^{(bs)}}{\bar{N}_\mu}$, hence the optimization problem can be recast as

$$\bar{D}^* = \min_{\mu} \frac{\bar{W}_\mu^{(s)} + \bar{W}_\mu^{(bs)}}{\bar{N}_\mu} \text{ s.t. } \bar{\mathcal{E}}_\mu \triangleq \bar{E}_\mu - P_{\text{avg}} \bar{T}_\mu \leq 0, \quad (7)$$

where $\bar{\mathcal{E}}_\mu = \bar{\mathbb{E}}_\mu[E_u - P_{\text{avg}} \Delta_u]$ is the *excess energy cost*. Note the inherent complexity to solve (7): as the policy varies, the delay metric changes both the numerator and denominator of the objective function, precluding a direct application of dynamic programming tools.

Alternative Problem Formulation: To address this challenge, we now devise a surrogate

optimization metric, by characterizing upper and lower bounds to \bar{D}_μ . To this end, let us define a "baseline" policy μ_{BS} as the one such that all requests are served by the BS and the UAV flies around at minimum power P_{min} (this policy is feasible). Since the delay to serve a request from a GN in position (r, θ) by direct transmission to the BS is $\frac{L}{R_{GB}(r)}$, the expected delay under policy μ_{BS} is obtained by computing the expectation with respect to the radial coordinate, $\bar{D}_{BS} \triangleq \int_0^a \frac{L}{R_{GB}(r)} f_R(r) dr$. Clearly, optimization of the policy yields $\bar{D}^* \leq \bar{D}_{BS}$. Under any policy μ (including μ^*) better than μ_{BS} (i.e., such that $\bar{D}_\mu \leq \bar{D}_{BS}$), the following bounds hold.

Proposition 2. Let μ be such that $\bar{D}_\mu \leq \bar{D}_{BS}$. Then, it holds that

$$\bar{W}_\mu^{(s)} \leq \bar{D}_\mu \leq \bar{W}_\mu^{(s)} \frac{1 + \Lambda' \bar{D}_{BS}}{1 + \Lambda' \bar{W}_\mu^{(s)}} \leq \bar{D}_{BS}. \quad (8)$$

Proof. See Appendix B. ■

Noticing that both the lower and upper bounds of \bar{D}_μ are increasing functions of $\bar{W}_\mu^{(s)}$, in our subsequent analyses we will focus on the alternative optimization problem

$$\min_\mu \bar{W}_\mu^{(s)} \text{ s.t. } \bar{\mathcal{E}}_\mu \leq 0. \quad (9)$$

In Sec. VI (see Table III), we show that this alternative formulation leads to a near-optimal solution with respect to the original optimization (6). To solve (9), we define the Lagrangian

$$g(\nu) = \min_\mu \bar{W}_\mu^{(s)} + \nu \bar{\mathcal{E}}_\mu = \min_\mu \lim_{t \rightarrow \infty} \mathbb{E}_\mu \left[\frac{1}{M_t} \sum_{u=1}^{M_t} (\Delta_u^{(s)} + \nu(E_u - P_{avg} \Delta_u)) \right], \quad (10)$$

where ν is the dual variable, optimized by solving $\max_{\nu \geq 0} g(\nu)$. We now demonstrate that for a given $\nu \geq 0$, (10) can be cast as a Semi-Markov Decision Process (SMDP) and solved with dynamic programming tools. Next, we discuss the SMDP states, actions, transitions, and policy.

States: The state is defined by the UAV position \mathbf{q}_U , an element of the set $\mathcal{Q}_{UAV} \triangleq \mathbb{R}_+ \times [0, 2\pi)$ (polar coordinates), and the position \mathbf{q}_G of the GN originating traffic, taking values from the set $\mathcal{Q}_{GN} \triangleq [0, a] \times [0, 2\pi)$. The state space is then $\mathcal{S} = \mathcal{S}_{\text{wait}} \cup \mathcal{S}_{\text{comm}}$, where $\mathcal{S}_{\text{wait}} = \mathcal{Q}_{UAV}$ is the set of *waiting* states and $\mathcal{S}_{\text{comm}} = \mathcal{Q}_{UAV} \times \mathcal{Q}_{GN}$ is the set of *communication* states. Crucial to the definition of the SMDP is how the system is sampled in time to define Markovian dynamics in the evolution of the sampled states: accordingly, we define the actions available in each state $s \in \mathcal{S}$ and the transition probabilities, along with the time duration $T(s; a)$, the UAV energy usage $E(s; a)$, and the request service delay $\Delta(s; a)$ metrics accrued in state s under action a .

Waiting states' actions, transitions, and metrics: In waiting state $\mathbf{s} = \mathbf{q}_U \in \mathcal{S}_{\text{wait}}$ at time t , i.e., the UAV is in position $\mathbf{q}_U(t) = \mathbf{q}_U = (r_U, \theta_U)$ with no active requests, then the UAV moves with radial and angular velocity components (v_r, θ_c) , over an arbitrarily small duration $\Delta_0 \ll \frac{1}{\Lambda'}$. Thus, the waiting-state action space is $\mathcal{A}_{\text{wait}}(r_U) \triangleq \left\{ (v_r, \theta_c) \in \mathbb{R}^2 \mid \sqrt{v_r^2 + r_U^2 \cdot \theta_c^2} \leq V_{\max} \right\}$, where $v_U = \sqrt{v_r^2 + r_U^2 \theta_c^2}$ is the velocity expressed using polar coordinates. Upon choosing action $\mathbf{a} = (v_r, \theta_c) \in \mathcal{A}_{\text{wait}}(r_U)$, the communication delay is $\Delta(\mathbf{s}; \mathbf{a}) = 0$, since there is no ongoing communication; the duration of a waiting state is $T(\mathbf{s}; \mathbf{a}) = \Delta_0$, and the UAV's energy use is $E(\mathbf{s}; \mathbf{a}) = \Delta_0 P_{\text{mob}}(v_U)$ to move at velocity v_U . The new state is then sampled at time $t + \Delta_0$, with the UAV moved to the new position $\mathbf{q}_U(t + \Delta_0) \approx (r_U, \theta_U) + (v_r, \theta_c) \Delta_0$. With probability $e^{-\Lambda' \Delta_0}$, no new request is received in the time interval $[t, t + \Delta_0]$, so that the new state is a waiting state. Otherwise, a new request is received from a GN in position (r, θ) (communication state). The transition probabilities from the waiting state $\mathbf{s}_n = \mathbf{q}_U \in \mathcal{S}_{\text{wait}}$ under action $\mathbf{a}_n = (v_r, \theta_c) \in \mathcal{A}_{\text{wait}}(r_U)$ are thus

$$\begin{aligned} \mathbb{P}(\mathbf{s}_{n+1} = \mathbf{q}_U + \mathbf{a}_n \Delta_0 | \mathbf{s}_n, \mathbf{a}_n) &= e^{-\Lambda' \Delta_0}, \\ \mathbb{P}(\mathbf{s}_{n+1} = (\mathbf{q}_U + \mathbf{a}_n \Delta_0, \mathbf{q}'_G) \text{ with } \mathbf{q}'_G \in \mathcal{F} | \mathbf{s}_n, \mathbf{a}_n) &= \frac{A(\mathcal{F})}{\pi a^2} \cdot (1 - e^{-\Lambda' \Delta_0}), \quad \forall \mathcal{F} \subseteq \mathcal{Q}_{\text{GN}}, \end{aligned} \quad (11)$$

where $A(\mathcal{F})$ is the area of region \mathcal{F} , since requests are uniformly distributed in the cell.

Communication states' actions, transitions, and metrics: Upon reaching a communication state $\mathbf{s}_n = (\mathbf{q}_U, \mathbf{q}_G) \in \mathcal{S}_{\text{comm}}$ at time t , the system must serve a GN request at position $\mathbf{q}_G = (r, \theta)$. The BS first determines the scheduling decision $\xi \in \{0, 1\}$. If $\xi = 0$, denoted as the action $\mathbf{a} = \text{BS}$, the GN transmits directly to the BS; the next state is the waiting state $\mathbf{s}_{n+1} = \mathbf{q}_U$, sampled immediately after, resulting in the energy-time metrics $E(\mathbf{s}_n; \mathbf{a}) = T(\mathbf{s}_n; \mathbf{a}) = 0$, and service delay metric $\Delta(\mathbf{s}_n; \mathbf{a}) = \frac{L}{\bar{R}_{GB}(r)}$ (time required to transmit the payload with throughput $\bar{R}_{GB}(r)$ between the GN and the BS). Instead, if $\xi = 1$, the UAV uses the D&F protocol, while following a trajectory starting from its current position \mathbf{q}_U and ending in position \mathbf{q}'_U . We denote this action as $\mathbf{a} = (\mathbf{q}_U \rightarrow \mathbf{q}'_U)$. In the *decode* phase of D&F (of duration t_p), the GN transmits its data payload to the UAV; in the *forward* phase (of duration $\Delta - t_p$), the UAV relays it to the BS. Assuming a *move-and-transmit* strategy [10], the trajectory $(\mathbf{q}_U \rightarrow \mathbf{q}'_U)$ and the durations (t_p and $\Delta - t_p$) must satisfy the data payload constraints (C.1), i.e., the entire payload of L bits is first transmitted to the UAV with throughput $\bar{R}_{GU}(r_{GU}(\eta))$, and then relayed to the BS with throughput $\bar{R}_{UB}(r_{UB}(\eta))$, where $r_{GU}(\eta)$ and $r_{UB}(\eta)$ are the GN-UAV and UAV-BS distances (projected onto the $x-y$ plane) at time η along the trajectory, respectively, so that the total communication delay is Δ .

For this action, the cost metrics are $\Delta(\mathbf{s}_n; \mathbf{a}) = T(\mathbf{s}_n; \mathbf{a}) = \Delta$ and $E(\mathbf{s}_n; \mathbf{a}) = \int_0^\Delta P_{\text{mob}}(v_U(\eta)) d\eta$. Upon completing D&F at time $t+\Delta$, the UAV enters the waiting state $(\mathbf{s}_{n+1} = \mathbf{q}'_U)$. The set of feasible UAV trajectories from \mathbf{q}_U to \mathbf{q}'_U , to serve a GN at position \mathbf{q}_G is

$$\mathcal{Q}_{\mathbf{q}_G}(\mathbf{q}_U \rightarrow \mathbf{q}'_U) \triangleq \left\{ \mathbf{p}_U : [0, \Delta] \mapsto \mathbb{R}_+ \times [0, 2\pi) \text{ s.t.} \right. \quad (12)$$

$$\int_0^{t_p} \bar{R}_{GU}(r_{GU}(\eta)) d\eta \geq L, \quad \int_{t_p}^\Delta \bar{R}_{UB}(r_{UB}(\eta)) d\eta \geq L, \quad (\text{C.1})$$

$$v_U(\eta) \leq V_{\max}, \quad \mathbf{p}_U(0) = \mathbf{q}_U, \quad \mathbf{p}_U(\Delta) = \mathbf{q}'_U, \quad \exists \Delta \geq 0, \exists 0 \leq t_p \leq \Delta \Big\}, \quad (\text{C.2})$$

where $v_U(\eta)$ is the UAV speed, C.1 reflects the data payload constraints, and C.2 the maximum speed and trajectory constraints. Then, the action space in state $(\mathbf{q}_U, \mathbf{q}_G) \in \mathcal{S}_{\text{comm}}$ when $\xi=1$ is the set $\mathcal{Q}_{\mathbf{q}_G}(\mathbf{q}_U) \triangleq \cup_{\mathbf{q}'_U \in \mathcal{Q}_{\text{UAV}}} \mathcal{Q}_{\mathbf{q}_G}(\mathbf{q}_U \rightarrow \mathbf{q}'_U)$ of feasible trajectories starting in \mathbf{q}_U that serve the GN at \mathbf{q}_G via the D&F protocol. The overall action space of this communication state is then $\mathcal{A}_{\text{comm}}(\mathbf{q}_U, \mathbf{q}_G) \triangleq \{\text{BS}\} \cup \{\mathcal{Q}_{\mathbf{q}_G}(\mathbf{q}_U)\}$, including the scheduling decision $\xi \in \{0, 1\}$.

Policy μ : For waiting states $\mathbf{q}_U \in \mathcal{S}_{\text{wait}}$, the policy $\mu(\mathbf{q}_U) \in \mathcal{A}_{\text{wait}}(r_U)$ selects a velocity (v_r, θ_c) from the respective action space. Likewise, for communication states $(\mathbf{q}_U, \mathbf{q}_G) \in \mathcal{S}_{\text{comm}}$, the policy selects the scheduling decision $\xi \in \{0, 1\}$ and if $\xi=1$, the trajectory followed in the D&F protocol, i.e., $\mu(\mathbf{q}_U, \mathbf{q}_G) \in \mathcal{Q}_{\mathbf{q}_G}(\mathbf{q}_U)$. With a stationary policy μ defined, the Lagrangian metric $L_\mu^{(\nu)} \triangleq \bar{W}_\mu^{(s)} + \nu \bar{\mathcal{E}}_\mu$ in (10) is reformulated using Little's Law [35] and is written as

$$L_\mu^{(\nu)} = \lim_{N \rightarrow \infty} \mathbb{E}_\mu \left[\frac{\frac{1}{N} \sum_{n=0}^{N-1} \ell_\nu(\mathbf{s}_n; \mu(\mathbf{s}_n))}{\frac{1}{N} \sum_{n=0}^{N-1} \mathbb{I}(\mathbf{s}_n \in \mathcal{S}_{\text{comm}})} \right] = \frac{1}{\pi_{\text{comm}}} \int_{\mathcal{S}} \Pi_\mu(\mathbf{s}) \ell_\nu(\mathbf{s}; \mu(\mathbf{s})) d\mathbf{s}, \quad (13)$$

where $\Pi_\mu(\mathbf{s})$ is the steady-state probability density function of being in state \mathbf{s} under policy μ , $\pi_{\text{comm}} = \int_{\mathcal{S}_{\text{comm}}} \Pi_\mu(\mathbf{s}) d\mathbf{s}$ is the steady-state probability that the UAV is in the communication phase, and $\ell_\nu(\mathbf{s}; \mathbf{a}) \triangleq \Delta(\mathbf{s}; \mathbf{a}) + \nu(E(\mathbf{s}; \mathbf{a}) - P_{\text{avg}} T(\mathbf{s}; \mathbf{a}))$ is the Lagrangian metric in state \mathbf{s} under action \mathbf{a} . In (13), $\sum_{n=0}^{N-1} \ell_\nu(\mathbf{s}_n; \mu(\mathbf{s}_n))$ is the total Lagrangian cost accrued during the first N SMDP stages, and $\sum_{n=0}^{N-1} \mathbb{I}(\mathbf{s}_n \in \mathcal{S}_{\text{comm}})$ is the number of communication states encountered; since a new decision interval initiates after a communication state, this equals the number of decision intervals (M_t in (10)). Taking the limit $N \rightarrow \infty$, $L_\mu^{(\nu)}$ is the expected Lagrangian cost per decision interval, as expressed in (10). The right-hand side expression in (13) follows because the SMDP reaches the steady-state when $N \rightarrow \infty$. Specializing, $\ell_\nu(r_U, \theta_U; v_r, \theta_c) = \nu(P_{\text{mob}}(\sqrt{v_r^2 + r_U^2 \theta_c^2}) - P_{\text{avg}}) \Delta_0$ for the waiting states, $\ell_\nu(r_U, \theta_U, r, \theta; \text{BS}) = \frac{L}{R_{GB}(r)}$ for direct-BS transmission in communication states, and $\ell_\nu(r_U, \theta_U, r, \theta; \mathbf{p}_U) = (1 - \nu P_{\text{avg}}) \Delta + \nu \int_0^\Delta P_{\text{mob}}(V(\eta)) d\eta$ for a communication relayed through

the UAV. The next proposition shows that the steady-state probability π_{comm} is independent of the policy μ , i.e., it is not affected by the optimization over μ .

Proposition 3. We have $\pi_{\text{comm}} = 1 - (2 - e^{-\Lambda' \Delta_0})^{-1}$.

Proof. See Appendix C. ■

This result permits rewriting (10) as an *average cost-per-stage problem*

$$g(\nu) = \frac{1}{\pi_{\text{comm}}} \min_{\mu} \int_S \Pi_{\mu}(s) \ell_{\nu}(s; \mu(s)) ds, \quad (14)$$

solvable through standard dynamic programming approaches (upon discretization of the state and action spaces), followed by the dual maximization $\max_{\nu \geq 0} g(\nu)$.

Two-stage policy decomposition: Since GN transmission requests are uniformly distributed in the circular cell, the UAV radius is a sufficient statistic in decision-making for a waiting state (r_U, θ_U) , expressed as $r_U \in \mathcal{S}_{\text{wait}} \triangleq [0, a]$. Likewise, for a communication state $(r_U, \theta_U, r, \theta)$, only the UAV radius, GN request radius, and the angle $\psi \in [0, 2\pi]$ between them suffice to characterize the state. Thus, communication states can be compactly represented as $(r_U, r, \psi = \theta - \theta_U) \in \mathcal{S}_{\text{comm}} \triangleq [0, a]^2 \times [0, 2\pi]$. Hence, the policy affects the SMDP state transitions (and its steady-state) only through the UAV radial velocity v_r in the waiting states, the scheduling decision (direct-BS or UAV relay) and UAV trajectory's end radius position \hat{r}_U in communication states. Instead, the angular velocity θ_c in the waiting states and the UAV trajectory to reach the target end radius \hat{r}_U in the communication states only affect the instantaneous Lagrangian ℓ_{ν} , but not state dynamics.

With this observation, let $O(r_U) \triangleq v_r \in [-V_{\max}, V_{\max}]$ define the radial velocity policy of waiting states $r_U \in \mathcal{S}_{\text{wait}}$, specifying the radial velocity component of waiting action $(v_r, \theta_c) \in \mathcal{A}_{\text{wait}}(r_U)$; let $U(r_U, r, \psi) \triangleq (\xi, \hat{r}_U)$ define the scheduling and next radius position policy of communication states $(r_U, r, \psi) \in \mathcal{S}_{\text{comm}}$: either direct-BS with $\hat{r}_U = r_U$ ($\xi = 0$), or any trajectory starting from radius r_U and ending at radius \hat{r}_U when relaying through the UAV ($\xi = 1$). Accordingly, O and U are the SMDP's *outer decisions* and are the only actions affecting the steady-state distribution, denoted as $\Pi_{O,U}$ under the outer policy (O, U) ; thus, (14) can be restated as

$$g(\nu) = \frac{1}{\pi_{\text{comm}}} \min_{O,U} \left[\int_{\mathcal{S}_{\text{wait}}} \Pi_{O,U}(\mathbf{s}) \ell_{\nu}^*(\mathbf{s}; O(\mathbf{s})) ds + \int_{\mathcal{S}_{\text{comm}}} \Pi_{O,U}(\mathbf{s}) \ell_{\nu}^*(\mathbf{s}; U(\mathbf{s})) ds \right], \quad (15)$$

where ℓ_{ν}^* is the Lagrangian metric optimized with respect to the *inner decision* components not specified by O and U . In particular, for a waiting state r_U , under the radial velocity action $O(r_U) = v_r$, the inner optimization is performed with respect to the angular velocity θ_c ,

$$\ell_\nu^*(r_U; v_r) = \min_{\theta_c} \nu (P_{\text{mob}}(V) - P_{\text{avg}}) \Delta_0 \quad \text{s.t. } V = \sqrt{v_r^2 + r_U^2 \theta_c^2} \leq V_{\max}. \quad (16)$$

Since $\nu \geq 0$, the optimizer θ_c^* is the angular velocity minimizing the UAV power consumption: due to the quasi-convex structure of $P_{\text{mob}}(v)$ [10], $\theta_c^* = 0$ if $|v_r| \geq v_{P_{\min}} \triangleq \arg \min_V P_{\text{mob}}(V)$ (in fact, any angular movement would undesirably increase power consumption), and $\sqrt{v_r^2 + r_U^2 (\theta_c^*)^2} = v_{P_{\min}}$ otherwise (i.e., enough angular movement to yield the power minimizing speed). For communication states, under direct-BS transmission, $\ell_\nu^*(\mathbf{s}; 0, r_U) = L/R_{GB}(r)$; on the other hand, when relaying through the UAV, ℓ_ν^* is obtained by optimizing the trajectory \mathbf{p}_U followed by the UAV, starting at radius r_U and terminating at radius \hat{r}_U (with final angular position $\hat{\phi}$ optimized),

$$\ell_\nu^*(\mathbf{s}; 1, \hat{r}_U) = \min_{\Delta, \mathbf{p}_U, t_p, \hat{\phi}} (1 - \nu P_{\text{avg}}) \Delta + \nu \int_0^\Delta P_{\text{mob}}(v_U(\eta)) d\eta \quad \text{s.t. C.1, C.2.} \quad (17)$$

where C.1-C.2 are the data payload, maximum UAV speed and trajectory constraints (see (12)). In other words, the inner decision on trajectory minimizes the instantaneous delay-energy trade-off, among all feasible trajectories terminating at the target radius \hat{r}_U . Defining $\alpha \triangleq \frac{\nu P_{\max}}{(1 + \nu(2P_{\max} - P_{\text{avg}}))} \in [0, 1]$ to regulate the trade-off between service delay and UAV energy, (17) can be rewritten as

$$\frac{\ell_\nu^*(\mathbf{s}; 1, \hat{r}_U)}{1 + \nu(2P_{\max} - P_{\text{avg}})} = \min_{\Delta, \mathbf{p}_U, t_p} (1 - 2\alpha) \Delta + \alpha \int_0^\Delta \frac{P_{\text{mob}}(V(\eta))}{P_{\max}} d\eta \quad \text{s.t. C.1, C.2,} \quad (18)$$

This reformulation is the focus of our HCSO trajectory design algorithm, detailed in Sec. IV.

Alg. 1 optimizes the outer policy and computes the average cost-per-stage metric $g(\nu)$, along with the average excess energy-per-stage metric for a given ν , by solving problem (15) via value iteration [25]. Alg. 2 solves the dual maximization $\max_{\nu \geq 0} g(\nu)$ via projected sub-gradient ascent¹ [36]. Specifically, in Alg. 1, lines 2 and 3 compute the inner Lagrangian cost metric optimized with respect to the inner actions—along with the excess energy cost metric—for all states and outer actions; line 6 computes the value iteration update for waiting states: upon moving to the new radial position $r_U + v_r \Delta_0$, no request is received, w.p. $e^{-\Lambda' \Delta_0}$, hence moving to a waiting state (with future value $V_{W,i}(r_U + v_r \Delta_0)$); otherwise, the system moves to a communication state, with future value $V_{C,i}(r_U + v_r \Delta_0)$ (averaged with respect to the request position); line 12 computes the value iteration update for communication states, transitioning to a waiting state w.p. 1; the corresponding optimal outer actions are saved in lines 7 and 13; line 16 averages the value of communication states with respect to the random request position; lines 8, 14, and 17 similarly

¹The source code for these algorithms is available on GitHub [2].

Algorithm 1 $(O^*, U^*, g(\nu), \bar{\mathcal{E}}, V_{\cdot,0}^{next}, \mathcal{E}_{\cdot,0}^{next}) = \text{VITER}(\nu, V_{\cdot,0}, \mathcal{E}_{\cdot,0})$

```

1: Initialization:  $i=0$ ; stop criterion  $\delta$ .
2: Inner optimization in waiting states:  $\forall r_U \in \mathcal{S}_{\text{wait}}, \forall v_r \in [-V_{\max}, V_{\max}]$ , calculate  $\ell_\nu^*(r_U; v_r)$  as in (16), with minimizer  $\theta_c^*$ ; compute excess energy cost  $\epsilon^*(r_U; v_r) = P_{\text{mob}}(\sqrt{v_r^2 + r_U^2(\theta_c^*)^2})\Delta_0 - P_{\text{avg}}\Delta_0$ .
3: Inner optimization in communication states:  $\forall s \in \mathcal{S}_{\text{comm}}, \forall \hat{r}_U \in [0, a]$ , calculate  $\ell_\nu^*(s; 1, \hat{r}_U)$  via Alg. 3 with  $\alpha = \nu P_{\max}/(1+\nu(2P_{\max}-P_{\text{avg}}))$ , with minimizer  $\mathbf{p}_U^*$  (trajectory); compute excess energy cost  $\epsilon^*(s; \hat{r}_U) = E(s; \mathbf{p}_U^*) - P_{\text{avg}}T(s; \mathbf{p}_U^*)$ .
4: repeat
5:   for each  $r_U \in [0, a]$  do ▷ Outer optimization in waiting states
6:      $V_{W,i+1}(r_U) \leftarrow \min_{v_r \in [-V_{\max}, V_{\max}]} [\ell_\nu^*(r_U; v_r) + e^{-\Lambda' \Delta_0} V_{W,i}(r_U + v_r \Delta_0) + (1 - e^{-\Lambda' \Delta_0}) V_{C,i}(r_U + v_r \Delta_0)]$ ,
7:      $O_{i+1}(r_U) \leftarrow v_r^*$ , where  $v_r^*$  is the arg min.
8:      $\mathcal{E}_{W,i+1}(r_U) \leftarrow \epsilon^*(r_U; v_r^*) + e^{-\Lambda' \Delta_0} \mathcal{E}_{W,i}(r_U + v_r^* \Delta_0) + (1 - e^{-\Lambda' \Delta_0}) \mathcal{E}_{C,i}(r_U + v_r^* \Delta_0)$ .
9:   end for
10:  for each  $r_U \in [0, a]$  do ▷ Outer optimization in communication states
11:    for each  $r \in [0, a], \psi \in [0, 2\pi] (s = (r_U, r, \psi))$  do ▷ Outer optimization in communication states
12:       $\hat{V}(s) \leftarrow \min \left\{ \underbrace{\frac{L}{R_{GB}(r)} + V_{W,i}(r_U)}_{\xi=0}, \underbrace{\min_{\hat{r}_U \in [0, a]} \ell_\nu^*(s; \hat{r}_U) + V_{W,i}(\hat{r}_U)}_{\xi=1} \right\}$  ▷ Value function given GN position
13:       $U_{i+1}(s) \leftarrow (\xi^*, \hat{r}_U^*)$ , where  $(\xi^*, \hat{r}_U^*)$  is the arg min ( $\hat{r}_U^* = r_U$  if  $\xi^* = 0$ ).
14:       $\hat{\mathcal{E}}(s) \leftarrow \xi^* \cdot \epsilon^*(s; \hat{r}_U^*) + \mathcal{E}_{W,i}(\hat{r}_U^*)$ . ▷ Total excess cost given GN pos., optimized over scheduling/trajecory
15:    end for
16:     $V_{C,i+1}(r_U) \leftarrow \int_0^{2\pi} \frac{1}{2\pi} \int_0^a \frac{2r}{a^2} \hat{V}(r_U, r, \psi) dr d\psi'$  ▷ Value function in comm states, averaged over GN position
17:     $\mathcal{E}_{C,i+1}(r_U) \leftarrow \int_0^{2\pi} \frac{1}{2\pi} \int_0^a \frac{2r}{a^2} \hat{\mathcal{E}}(r_U, r, \psi) dr d\psi'$  ▷ Excess energy cost in comm states, averaged over GN position
18:  end for
19:   $\forall r_U \in [0, a]$  and  $X \in \{W, C\}$ , calculate  $\delta_X^{(V)}(r_U) = V_{X,i+1}(r_U) - V_{X,i}(r_U)$  and  $\delta_X^{(\mathcal{E})}(r_U) = \mathcal{E}_{X,i+1}(r_U) - \mathcal{E}_{X,i}(r_U)$ ;  $i \leftarrow i+1$ .
20: until  $\max_{r_U, X} \delta_X^{(V)}(r_U) - \min_{r_U, X} \delta_X^{(V)}(r_U) < \delta$  and  $\max_{r_U, X} \delta_X^{(\mathcal{E})}(r_U) - \min_{r_U, X} \delta_X^{(\mathcal{E})}(r_U) < \delta$ . ▷ Termination condition
21: return  $g(\nu) \approx \delta_W^{(V)}(0)/\pi_{\text{comm}}$ ,  $\bar{\mathcal{E}} \approx \delta_W^{(\mathcal{E})}(0)$ . ▷ dual cost and average excess energy cost
22:  $V_{\cdot,0}^{next}(\cdot) = V_{\cdot,0}(\cdot) - V_{W,i}(0)$ ,  $\mathcal{E}_{\cdot,0}^{next}(\cdot) = \mathcal{E}_{\cdot,0}(\cdot) - \mathcal{E}_{W,i}(0)$ . ▷ Relative values (next VITER initialization)
23:  $O^*(\cdot) = O_i(\cdot)$ ,  $U^*(\cdot) = U_i(\cdot)$ . ▷ Optimal waiting and communication policies

```

update the total excess energy cost, needed to compute the projected dual sub-gradient ascent in Alg. 2. In practice, the integrals in lines 16 and 17, and the continuous state/action spaces are discretized (see MAESTRO-X [2]), leading to an overall complexity of each value iteration update (lines 5-18) of order $\mathcal{O}(K_R \cdot (K_V + K_R^2 \cdot K_A))$, where K_R is the number of discretized radii levels (r_U and r values), K_A is the number of angular levels (ψ and ψ'), and K_V is the number of discretized radial velocities (v_r). Upon convergence (typically, value iteration converges within $\mathcal{O}(\log(1/\delta))$ iterations to achieve a target accuracy δ [25, Sec. V]), line 21 estimates the values of the average cost-per-stage and excess energy-per-stage metrics.

In Alg. 2, line 1 initializes the dual variable and a sequence of step-sizes used for projected sub-gradient ascent; line 3 calls value iteration (Alg. 1) using the current dual variable ν , and outputs the optimal outer policy and the average cost-, excess energy- per-stage metrics; line 5 monitors convergence in terms of primal feasibility and complementary slackness conditions; line 4 updates the value of the dual variable in the direction of its sub-gradient and projects its value to the non-negative range to ensure dual feasibility; note that Alg. 1 outputs also the *relative values* metrics V and \mathcal{E} : these are used to initialize the total cost and excess energy metrics in the next call to Alg. 1, and help speed up convergence. We are left with the trajectory design (line 3 of Alg. 1), carried out using Hierarchical CSO in the next section.

Algorithm 2 Projected Sub-gradient Ascent (PSGA)

```

1: Initialization:  $k = 0$ ; dual variable  $\nu \geq 0$ ; step-size  $\{\rho_k = \frac{\rho_0}{k+1}, k \geq 0\}$ ;  $V_{\cdot,0}(\cdot) = \mathcal{E}_{\cdot,0}(\cdot) \equiv 0$ .
2: repeat
3:    $(O^*, U^*, g, \bar{\mathcal{E}}, V_{\cdot,0}, \mathcal{E}_{\cdot,0}) \leftarrow \text{VITER}(\nu, V_{\cdot,0}, \mathcal{E}_{\cdot,0})$  via Alg. 1.
4:   Update  $\nu \leftarrow \max\{\nu + \rho_k \bar{\mathcal{E}}, 0\}$ ;  $k \leftarrow k+1$ . ▷ Dual variable update
5: until  $\bar{\mathcal{E}} < \epsilon_{PF}$ ;  $\nu |\bar{\mathcal{E}}| < \epsilon_{CS}$  ▷ Check KKT optimality conditions
6: return: optimal outer policy  $(O^*, U^*)$ .

```

Algorithm 3 HCSO Algorithm

```

1: Randomly initialize  $N$  particles  $(\mathbf{p}, \mathbf{v})_{1:N}$ :  $\mathbf{p}_i$  is a sequence of way-points,  $\mathbf{v}_i$  a sequence of UAV speeds.
2: while  $M \leq M_{\max}$  do
3:   Obtain  $M$ -segment trajectory:  $(\mathbf{p}^*, \mathbf{v}^*) = \text{CSO}(\mathbf{p}_{1:N}, \mathbf{v}_{1:N}, N, M)$  (see [26]). ▷ CSO call
4:   Increase  $M \leftarrow 2M$ ; interpolate to form reference trajectory:  $(\tilde{\mathbf{p}}, \tilde{\mathbf{v}}) = \text{interp}(\mathbf{p}^*, \mathbf{v}^*, M)$ . ▷ Increase resolution via interpolation
5:   Reduce swarm size  $N \leftarrow N - N_{\text{red}}$ .
6:   for  $n=1, 2, \dots, N$  do ▷ Generate  $N$  particles randomly
7:     New way-point particle  $\mathbf{p}_n$  with  $m$ th way-point  $\mathbf{x}_m = \tilde{\mathbf{x}}_m + (\chi_m, \zeta_m)$  and  $\mathbf{x}_M = \hat{r}_U \frac{\mathbf{x}_{M-1}}{\|\mathbf{x}_{M-1}\|_2}$ . ▷ Way-point perturbation
8:     New velocity particle  $\mathbf{v}_n$  with  $m$ th velocity  $v_m = [\tilde{v}_m + \varkappa_m]^{[V_{\text{low}}, V_{\text{max}}]}$ . ▷ Velocity perturbation
9:   end for
10:  end while

```

IV. TRAJECTORY DESIGN VIA HIERARCHICAL COMPETITIVE SWARM OPTIMIZATION

In this section, we design the UAV trajectory during the D&F protocol. To solve (18), we propose a CSO scheme [26] defining a *meta-heuristic UAV trajectory*. First, as done also with SCA approaches [10], [16], [37], we simplify the continuous UAV trajectory into a finite sequence of way-points connected by straight lines at constant velocity. However, a direct application of CSO to high-resolution trajectory design suffers from poor convergence due to exponentially large solution spaces [38]. We address this weakness by proposing a Hierarchical variant of CSO (HCSO), wherein a sequence of problems is solved: initially, CSO produces a low-resolution trajectory; the optimized trajectory is then interpolated to create a higher-resolution one, then further optimized with CSO. The process repeats until a target resolution is achieved.

Let $\mathbf{x}_0 = (r_U, 0)$ be the initial UAV position and $\mathbf{x}_G \triangleq (r \cos \psi, r \sin \psi)$ be the request position (in this section, expressed as Cartesian coordinates), corresponding to the communication state $\mathbf{s} = (r_U, r, \psi) \in \mathcal{S}_{\text{comm}}$. Given a target end radius position \hat{r}_U (the outer action), we encode the UAV trajectory as a sequence of M way-points $\mathbf{x}_m = (x_m, y_m)$, $m = 1, \dots, M$, ending at \mathbf{x}_M at radius \hat{r}_U , and velocities $v_m \in [V_{\text{low}}, V_{\text{max}}]$ used to traverse each straight trajectory segment $\Psi_m \triangleq \mathbf{x}_m - \mathbf{x}_{m-1}$. The first and second $\frac{M}{2}$ segments correspond to the two phases of the D&F protocol. Here, the minimum velocity $V_{\text{low}} \ll V_{\text{max}}$ ensures well-defined segment durations; the sequences of way-points $\mathbf{p} \triangleq [\mathbf{x}_1, \dots, \mathbf{x}_M]$ and velocities $\mathbf{v} \triangleq [v_1, \dots, v_M]$ are the optimization variables. Since the number of bits communicated (C.1) during each trajectory segment, coupled with our throughput-maximizing rate adaptation scheme, cannot be computed in closed-form, we approximate them numerically. Specifically, between subsequent way-points \mathbf{x}_{m-1} and \mathbf{x}_m

traversed with velocity v_m , we generate a sequence of n_{res} evenly-spaced points with sufficiently high resolution; letting $\{R_k^{\text{new}}\}_{k=1}^{n_{\text{res}}}$ be the expected throughput at each point, computed via (3) and Prop. 1, the number of bits communicated along the m th segment is approximated as $F_m \triangleq \frac{\|\Psi_m\|_2}{v_m} \frac{1}{n_{\text{res}}} \sum_{k=1}^{n_{\text{res}}} R_k^{\text{new}}$, where $\frac{\|\Psi_m\|_2}{v_m}$ is the time taken to traverse it. Thus, (18) becomes

$$(\mathbf{P.0}) \quad \min_{\mathbf{p}, \mathbf{v} \in [V_{\text{low}}, V_{\text{max}}]^M} \sum_{m=1}^M \frac{\|\Psi_m\|_2}{v_m} \left(1 - 2\alpha + \alpha \frac{P_{\text{mob}}(v_m)}{P_{\text{max}}}\right) \quad (19)$$

$$\text{s.t. } h_i(\mathbf{p}, \mathbf{v}) \triangleq L - \sum_{m=\frac{M}{2}i+1}^{\frac{M}{2}(i+1)} F_m \leq 0, \quad i = 0 \text{ and } 1, \|\mathbf{x}_M\|_2 = \hat{r}_U, \quad (\tilde{\mathbf{C}})$$

where $\tilde{\mathbf{C}}$ enforce the data payload and end radius constraints. To solve (P.0) with CSO, we first convert it into an unconstrained one, by penalizing constraint violations with a particular solution: 1) if the UAV does not decode (or forward) its data payload by the end of either phase, then it flies along the circumference of a circle (radius $r_{\min} > 0$, small) around the current position with its power-minimizing velocity ($v_{P_{\min}} = 22$ m/s [10]) until the transmission/reception is completed; and 2) we enforce the end radius constraint by projecting the penultimate way-point \mathbf{x}_{M-1} to the circle at radius \hat{r}_U , i.e. $\mathbf{x}_M = \hat{r}_U \mathbf{x}_{M-1} / \|\mathbf{x}_{M-1}\|_2$.² This yields the penalized objective function

$$\begin{aligned} \hat{f}(\mathbf{p}, \mathbf{v}) &\triangleq \sum_{m=1}^M \frac{\|\Psi_m\|_2}{v_m} \left(1 - 2\alpha + \alpha \frac{P_{\text{mob}}(v_m)}{P_{\text{max}}}\right) + (1 - 2\alpha)(\hat{t}_{P,0} + \hat{t}_{P,1}) + \alpha \frac{\hat{E}_{P,0} + \hat{E}_{P,1}}{P_{\text{max}}}; \\ \hat{t}_{P,0} &\triangleq \frac{\max\{h_0(\mathbf{p}, \mathbf{v}), 0\}}{\bar{R}_{GU}(\|\mathbf{x}_{M/2} - \mathbf{x}_G\|_2)}; \quad \hat{t}_{P,1} \triangleq \frac{\max\{h_1(\mathbf{p}, \mathbf{v}), 0\}}{\bar{R}_{UB}(\|\mathbf{x}_M\|_2)}; \quad \hat{E}_{P,i} \triangleq P_{\min} \hat{t}_{P,i}, \quad \mathbf{x}_M = \hat{r}_U \frac{\mathbf{x}_{M-1}}{\|\mathbf{x}_{M-1}\|_2}, \end{aligned}$$

where $\hat{t}_{P,i}$ and $\hat{E}_{P,i}$ are the time and energy penalties involved in finishing the data communication during the decode and forward phases ($i=0$ and 1). In particular, $\hat{t}_{P,i}$ equals the remaining payload $\max\{h_i(\mathbf{p}, \mathbf{v}), 0\}$, divided by the corresponding throughput at the terminal position (\bar{R}_{GU} for the decode phase and \bar{R}_{UB} for the forward phase). Hence, (P.0) becomes $\min_{\mathbf{p}, \mathbf{v}} \hat{f}(\mathbf{p}, \mathbf{v})$. To solve this problem, we employ the HCSO algorithm, outlined in Alg. 3 and discussed next.

We initialize N way-point particles $\mathbf{p}_{1:N} \triangleq \mathbf{p}_1, \dots, \mathbf{p}_N$ and N UAV velocity particles $\mathbf{v}_{1:N} \triangleq \mathbf{v}_1, \dots, \mathbf{v}_N$ (line 1). The core of the algorithm is CSO (line 3), detailed in [26]: essentially, during the k th iteration within CSO, the N particles are randomly grouped into $\frac{N}{2}$ pairwise competitions. For both members of a pair, $\hat{f}(\mathbf{p}, \mathbf{v})$ is calculated; the winner of the competition is passed onto the $(k+1)$ th iteration, while the loser is modified by learning from the winner,

²We let $\frac{\mathbf{x}}{\|\mathbf{x}\|_2} = (1, 0)$ for a point in the origin, $\mathbf{x} = (0, 0)$.

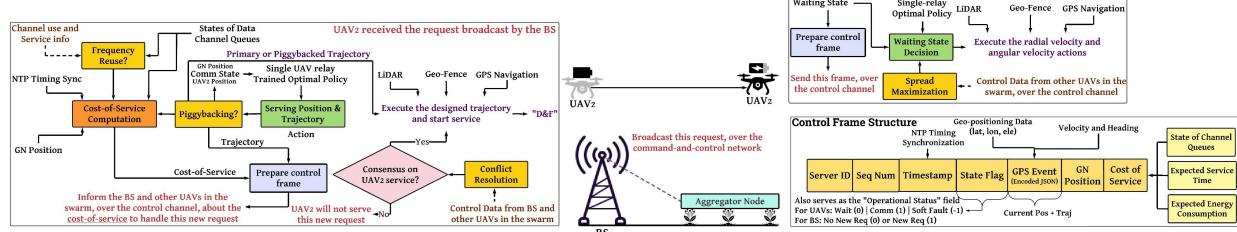


Fig. 3: An illustration outlining the sequence of operations under MAESTRO-X that occur at each UAV.

as detailed by the update equations in [26]; after repeating these pair-wise competitions, the CSO algorithm outputs a winning trajectory ($\mathbf{p}^*, \mathbf{v}^*$). However, a direct application of CSO alone suffers from a complexity-accuracy dilemma: high-resolution trajectories are slow to converge, while low-resolution ones give rise to poor solutions that fail to capture fine-grained variations in the trajectory way-points and velocities. To overcome this limitation, we embed CSO within a hierarchical wrapper: starting from a low-resolution trajectory optimized via CSO, after each CSO iteration (line 3), the resulting trajectory is interpolated to form a reference higher-resolution trajectory of $M \leftarrow 2M$ way-points (line 4). The new population size is then reduced, $N \leftarrow N - N_{\text{red}}$, to lower the computational burden of CSO (line 5), and a new set of N particles is generated randomly. **To preserve the quality of the previous lower-resolution trajectory solution**, the m th way-point of each new particle is generated by injecting zero-mean Gaussian noise $\chi_m, \zeta_m \sim \mathcal{N}(0, \sigma_{m,X}^2)$ (line 7) around the reference trajectory; similarly, the UAV velocity is generated by injecting Gaussian noise $\varkappa_m \sim \mathcal{N}(0, \sigma_V^2)$ (line 8), followed by projection onto the feasible set $([.]^{[V_{\text{low}}, V_{\text{max}}]})$. Here, the way-point variance $\sigma_{m,X}^2 = \varsigma(\|\tilde{\mathbf{x}}_{m+1} - \tilde{\mathbf{x}}_m\|^2 + \|\tilde{\mathbf{x}}_{m-1} - \tilde{\mathbf{x}}_m\|^2)$, with scaling factor $\varsigma > 0$, is determined by the spread between neighboring reference trajectory way-points. This choice accounts for the empirical observation that in areas with clustered UAV way-points, the objective function $\hat{f}(\mathbf{p}, \mathbf{v})$ is sensitive to large variations. The speed variance $\sigma_V^2 = \varepsilon(V_{\text{max}} - V_{\text{low}})^2$, with scaling factor $\varepsilon > 0$, reflects the observation that the UAV velocities exhibit faster convergence with CSO than the trajectory way-points and less sensitivity to random initialization. These steps in Alg. 3 continue until the desired trajectory resolution is reached.

V. MAESTRO-X: AN EXTENSION TO UAV SWARMS

In this section, we extend MAESTRO to swarms of N_U UAV-relays. This eXtension, termed MAESTRO-X, augments the multiscale optimal policy obtained via SMDP value iteration. Depicting an example scenario of serving data traffic generated by an aggregation of soil sensors in precision agriculture, Fig. 3 illustrates its control flow. **MAESTRO-X is enabled by replicating the optimal single-agent policy of the SMDP in Sec. III across the swarm and**

employing additional enhancements including *spread maximization*, *consensus-driven conflict resolution* with queuing dynamics, *piggybacking*, and *frequency reuse*. These mechanisms³ are implemented using a fully-connected distributed mesh network overlaid on the BS and UAVs, that enables periodic exchanges of command-and-control messages, as depicted in Fig. 3.

Spread Maximization: Note that the inner action of MAESTRO’s optimal waiting policy is symmetric in relation to clockwise and counter-clockwise angular UAV movements. For multiple UAVs, we leverage this symmetry to proactively position idle UAVs for potential new relay requests. Specifically, each UAV in the waiting state moves either clockwise or counter-clockwise (with angular velocity given by (16)), so as to maximize its angular distance from the nearest UAV in the waiting state, in an attempt to spread out and more readily serve future requests. To this end, UAV i parses the state flag as 0 and GPS event fields in its control frame (see Fig. 3). By monitoring the control frames received from other UAVs, it constructs a local peer list \mathcal{L} of other waiting state UAVs, and determines its closest peer (in the angular dimension) $j^* = \arg \min_{j \in \mathcal{L}} |\theta_i - \theta_j|$, where θ_j is the current angular coordinate of UAV j . UAV i then executes the angular motion away from UAV j^* , until new control frames (containing updated positions) are received from its peers (at the end of the synchronized reporting period) or upon receiving a new GN transmission request, at which time it transitions to the communication state.

Consensus-driven Conflict Resolution: In our single-UAV formulation (Sec. III), the scheduling action was determined by comparing the Lagrangian costs of direct-BS transmission to that of relayed UAV service. To extend scheduling decisions to UAV swarms—including queueing dynamics, as well as simultaneous multi-user service via piggybacking at the UAVs and frequency reuse (both described later in this section)—the augmented scheduling decision must now 1) resolve conflicts among the BS and UAVs as to whom should serve a new GN request; 2) facilitate a consensus on the best node to serve the GN; 3) account for queueing delays experienced at each potential server node while waiting for data channels to become available. Similarly to the single-UAV setting, this augmentation is driven by a cost-of-service metric computed at the BS and at each UAV. The new metric consists of several modifications to the original delay-energy cost trade-off computed in the single-UAV setting. For new requests served directly by the BS, the new metric equals the original delay metric, plus an estimate of the time needed for a data channel to become available (and considers the frequency reuse mechanism to be described).

³Due to space constraints, we keep our discussions on these multi-agent mechanisms brief. For more details on their implementation, please refer to our source code on GitHub [2].

This time can be estimated based on the time needed to complete the requests currently served at the BS, and the time needed to complete those already queued. Thus, for a new GN request at (r, θ) , the augmented cost metric associated with direct-BS transmission is $\frac{L}{R_{GB}(r)} + t_{BS}$, where the first term accounts for the transmission time, whereas t_{BS} is the additional waiting time.

Meanwhile, for new requests served by UAV i at radius $r_{U|i}$, GN request radius r , and angle between them $\psi_{U|i}$, i.e., state $\mathbf{s}_i = (r_{U|i}, r, \psi_{U|i})$, with target end radius $\hat{r}_{U|i}$, the augmented cost metric is given by $\tilde{\ell}_\nu^*(\mathbf{s}_i; 1, \hat{r}_{U|i}) + t_{U|i}$. The first term, $\tilde{\ell}_\nu^*(\mathbf{s}_i; 1, \hat{r}_{U|i})$, is the Lagrangian cost metric, modified to account for the piggybacking mechanism (described later in this section), wherein the UAV follows a collated trajectory to handle the new request while serving previous requests; the second term, $t_{U|i}$, is an estimate of the time needed for a data channel to become available (and considering the frequency reuse mechanism). Upon calculating these cost-of-service metrics for the BS and the UAVs, the network arrives at a consensus on the best node to serve the new request, i.e., if $\frac{L}{R_{GB}(r)} + t_{BS} \leq \tilde{\ell}_\nu^*(\mathbf{s}_i; 1, \hat{r}_{U|i}) + t_{U|i}, \forall i \in \{1, 2, \dots, N_U\}$, then the BS serves the request; otherwise, the request is relayed through the UAV $i^* = \arg \min_{i \in \{1, 2, \dots, N_U\}} \tilde{\ell}_\nu^*(\mathbf{s}_i; 1, \hat{r}_{U|i}) + t_{U|i}$.

Frequency Reuse: To improve the spectrum utilization efficiency, we propose a frequency reuse mechanism, allowing multiple serving nodes (the BS and UAVs) to share the same data channel simultaneously when serving their respective GN requests. When direct-BS transmission is used to serve a new GN request, a single data channel assignment occurs at the start of direct transmission. When the new request is instead served using a D&F UAV relay, two distinct data channel assignments occur: one each for the decode and forward phases of the UAV. In essence, reuse of an occupied data channel is permitted on the condition that the received SNRs of nodes sharing the data channel degrade no more than an acceptable pre-specified threshold permits. Moreover, to make operations of the frequency reuse mechanism more amenable to our problem, which includes UAVs following time-varying trajectories, we equivalently describe this SNR degradation threshold by instead using a minimum distance threshold d_{th} .

The frequency reuse mechanism proceeds in the same way, regardless of whether the data channel assignment under consideration is for a GN using direct-BS transmission, a GN sending its data to a UAV (decode phase), or a UAV relaying its data payload to the BS (forward phase). To formalize, let $k \in \{1, 2, \dots, N_C\}$ be the data channel under consideration for reuse; let node i be the new transmitter (either a GN beginning its uplink transmission or a UAV beginning its forward phase) determining whether reuse of data channel k is possible; let node j be the intended receiver of the transmission originating from node i ; let $\mathcal{T}(k)$ be the set of active

transmitters already using data channel k to serve their requests, i.e., a GN transmitting to a BS or UAV, or a UAV transmitting to the BS during its forward phase; let $\mathcal{R}(k)$ be the set of active receivers already using data channel k , i.e., a UAV receiving an uplink transmission from a GN during the decode phase, the BS receiving an uplink transmission directly from a GN, or the BS receiving the data payload from a UAV during the forward phase. For data channel k to be deemed acceptable for reuse, the following two conditions must both be met:

$$(\textbf{FR.1}) \quad d_{\ell,j} \geq d_{\text{th}}, \quad \forall \ell \in \mathcal{T}(k), \quad (20)$$

$$(\textbf{FR.2}) \quad d_{i,\ell} \geq d_{\text{th}}, \quad \forall \ell \in \mathcal{R}(k), \quad (21)$$

where $d_{i',j'}$ is the Euclidean distance between any transmitter i' and receiver j' . From the above equations, **(FR.1)** ensures that the distances between the intended receiver and all currently active transmitters are above the minimum distance threshold d_{th} , at all times during the execution of the UAVs' trajectories. Likewise, **(FR.2)** ensures that distances between the new transmitter and all currently active receivers are above the minimum distance threshold d_{th} . Effectively, satisfying conditions **(FR.1)** and **(FR.2)** simultaneously ensure that no received SNR experiences a degradation beyond a pre-specified limit, and hence data channel k is acceptable for reuse. Next, given its re-usability, the wait time for a channel to become available is estimated by modeling queuing dynamics, choosing the channel with the smallest wait time for service. Also, note that, once a channel is chosen with reuse, since the throughput experienced by the UAV during service degrades due to the added interference from other transmitters using the same channel, the UAV might not be able to complete its decode or forward phases using the optimal trajectory: the UAV then flies along the circumference of a circle ($r_{\min} > 0$) around the phase-specific final way-point with its power-minimizing velocity (22 m/s) to complete the phase; additionally, we evaluate the service in this case using the same time and energy penalties discussed in Sec. IV.

Piggybacking: To facilitate simultaneous multi-user service at the UAVs, we incorporate a piggybacking mechanism (in the cost-of-service computation of the consensus-driven conflict resolution process), wherein a UAV follows a collated trajectory to accommodate new GN uplink requests while serving previous requests. Recalling from the description of conflict resolution, for a new request served through UAV i , we consider the state $\mathbf{s}_i = (r_{U|i}, r, \psi_{U|i})$, with target end radius $\hat{r}_{U|i}$, and modified Lagrangian cost metric $\tilde{\ell}_\nu^*(\mathbf{s}_i; 1, \hat{r}_{U|i})$. If UAV i is currently not serving any other request, this modified cost metric simplifies to $\tilde{\ell}_\nu^*(\mathbf{s}_i; 1, \hat{r}_{U|i}) = \ell_\nu^*(\mathbf{s}_i; 1, \hat{r}_{U|i})$,

Notation	Description	Simulation Value	Notation	Description	Simulation Value
N_G	Number of GNs	30	a	Cell radius	1 km
L	Data payload	10 Mbits	W	System BW	20 MHz
N_C	Number of data channels	4	B	Data channel BW	5 MHz
κ	NLoS attenuation constant	0.2		SNR referenced at 1 m	40 dB
$(\alpha, \tilde{\alpha})$	LoS/NLoS pathloss exponents	(2,2.8)		UAV mobility power consumption	Eq. (4), params. of [10]
(k_1, k_2)	Rician K -factor parameters [14]	(1,0.05)	(z_1, z_2)	LoS probability parameters [39]	(9.61,0.16)
H_U / H_B	UAV / BS antenna height	200 m / 80 m	V_{\max}	Max. UAV speed	55 m/s
	Control frame reporting period	10 ms		SINR degradation threshold	5 dB

TABLE II: The system simulation parameters (unless otherwise stated).

i.e., the original Lagrangian cost metric computed for the single UAV. On the other hand, if the UAV is currently serving other requests, the UAV computes the cost metric to serve the new request by *piggybacking* it, i.e., serving it simultaneously with its current requests on a different data channel. In this case, the modified cost metric becomes $\tilde{\ell}_\nu^*(\mathbf{s}_i; 1, \hat{r}_{U|i}) = \ell_\nu^{(\text{pg})}(\mathbf{s}_i; 1, \hat{r}_{U|i})$, where $\ell_\nu^{(\text{pg})}(\mathbf{s}_i; 1, \hat{r}_{U|i})$ is defined to encapsulate modifications to the cost-of-service metric corresponding to the amount of data payload of the new request that has been either decoded or forwarded (or both) during the execution of the current trajectory (serving the UAV's previous requests). Note that the energy expended by the UAV serving its current trajectory while piggybacking the new request is not considered in the cost computed for this new request, since the energy cost has already been accounted for in the execution of the current trajectory; instead, we consider only the delays experienced by the piggybacked GN during its associated cost computation.

VI. SIMULATION SETUP AND EVALUATIONS

Unless otherwise stated, we use the parameter values in Table II. To solve (15) via Algorithms 1–3, we discretize the SMDP state and action spaces (with 25 equally-spaced radii levels and 25 radial velocity waiting actions) and apply linearly-interpolated value iteration (see implementation details documented in [2]). Furthermore, we chose $\Delta_0 = 1$ s.

Validation of surrogate optimization problem (9): First, we justify the efficacy of our alternative optimization framework that replaces the original metric \bar{D}_μ with the lower bound $\bar{W}_\mu^{(s)}$. As depicted in Table III, we observe that the optimized value $\bar{W}_{\mu^*}^{(s)}$ of the alternative formulation (9) is practically identical to the expected delay metric \bar{D}_{μ^*} of the original formulation (6), across various data payload sizes (L) and data traffic arrival rates (Λ'). Hence, replacing \bar{D}_μ with its lower bound $\bar{W}_\mu^{(s)}$ as the optimization metric leads to near-optimal solutions. Notably, the surrogate optimization problem (9) is amenable to dynamic programming tools such as value iteration (see Alg. 1) and enables our proposed two-scale policy decomposition that drastically reduces the size of the action space in our SMDP formulation. These tools would not be directly applicable to the original formulation (6) that uses \bar{D}_μ as the optimization objective.

Payload: L	Arrival rate: Λ'	Lower bound: $\bar{W}_{\mu^*}^{(s)}$	Expected Delay: \bar{D}_{μ^*}	Direct-to-BS: \bar{D}_{BS}
1 Mbit	1 req/min/UAV	1.15 s	1.15 s	31.64s
10 Mbit	0.2 req/min/UAV	16.41 s	16.41 s	316.38 s
100 Mbit	0.033 req/min/UAV	82.17 s	82.17 s	3163.81 s

TABLE III: $P_{avg}=1$ kW: A comparison between the lower bound $\bar{W}_{\mu^*}^{(s)}$ of \bar{D}_{μ^*} (Prop. 2) and direct-BS (\bar{D}_{BS}).

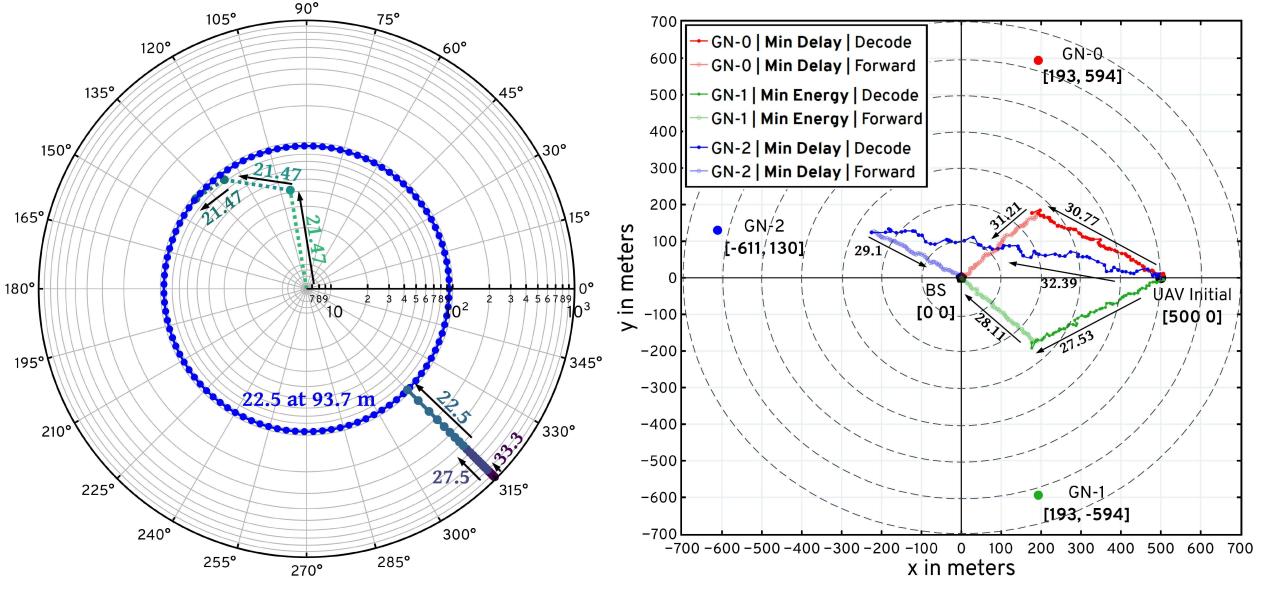


Fig. 4: $L=10$ Mbit, $P_{avg}=1.2$ kW, $\Lambda'=0.2$ req/min/UAV: Optimal waiting phase (a) and optimized D&F trajectory during a communication phase (terminating above the BS) (b). The arrows and associated numerical values represent the direction of motion and the flying speed in m/s.

MAESTRO policy: We now study illustrative examples of the optimal policy (Fig. 4). We note that, during the waiting phase (Fig. 4a), the UAV moves towards a radius of ≈ 94 m; upon reaching it, it flies at power-minimizing speed (22.5 m/s) along a circle: this allows the UAV to be well-positioned for future requests (not too close to the BS, and not too far away from it), and at the same time to minimize its power consumption. Next, Fig. 4b depicts the optimal trajectory obtained via HCSO (Algorithm 3), for a certain configuration of GN request positions, initial and target final UAV radii (evident from the figure). Intuitively, during the decode phase, the UAV flies towards the GN to improve the pathloss conditions; for the same reason, it moves towards the BS during the forward phase. Additionally, Fig 4b depicts two different trajectory choices for the GNs at $[193, \pm 594]$ m (GN-0 and GN-1, specular to each other), one corresponding to minimum service delay and the other corresponding to minimum service energy: here, in addition to observing the angular symmetry in our formulation (see Sec. III), we notice that, under the minimum delay trajectory, the UAV flies faster, to improve pathloss quicker and reduce

the transmission delay; in contrast, it flies slower under the minimum energy trajectory, to save energy. The delay-energy trade-off in trajectory design is regulated via α , as described by (18).

MAESTRO-X delay-power trade-off: We compare the delay-power trade-off of MAESTRO-X with adaptations of state-of-the-art algorithms to our setup, namely: the *CIRCLE* heuristic [20]; a CVXPY implementation of the Successive Convex Approximation scheme (*SCA*) [10]; a CVXPY implementation of the Constrained SCA scheme with Alternating Direction Method of Multipliers (*CSCA-ADMM*) [16], and a TensorFlow implementation of the Double Deep-Q Networks framework (*DDQN*) [19]. Note that all these frameworks are optimized under their original channel and communication models detailed in the corresponding references (see Table I for a list of their features), while we evaluate their performance under more realistic models of dynamic traffic arrivals and A2G channels. In addition, we consider the following custom heuristics: *BS-only*, in which GNs transmit directly to the BS without using UAVs; *HAP-only* in which GNs transmit directly to a High Altitude Platform (HAP, height=2 km); and *Static*, in which the UAVs statically hover at fixed locations. We also compute a *Lower Bound* to the delay as follows: for a GN at radius level r , it is the minimum between the delay incurred with direct-BS transmission (with throughput $\bar{R}_{GB}(r)$), and a D&F scheme in which the UAV is on top of the GN during the decode phase (with throughput $\bar{R}_{GU}(0)$), and on top of the BS during the forward phase (with throughput $\bar{R}_{UB}(0)$). Note that this lower bound is not attainable, since it neglects the mobility of the UAV. We average the results over 1000 requests.

In Fig. 5a, we plot the delay-power trade-off under low congestion ($\Lambda'=0.2$ req/min/UAV). Remarkably, MAESTRO-X allows to regulate the delay-power trade-off, whereas the other schemes do not. Across such trade-off, it outperforms all other schemes. Specifically, exploiting the mobility and maneuverability of the UAVs via optimized trajectories demonstrate lower service delays compared to static UAV deployments: for instance, a single UAV optimized via MAESTRO under 1 kW power constraint delivers the data payload 29% faster than a static UAV, while using 27% less power. Notably, under the same power consumption as the competitors, a single UAV optimized with MAESTRO achieves 38% lower delay than 3 UAV relays under DDQN [19], and 13 \times faster service times than the CIRCLE heuristic with 3 UAVs [20]. Adding UAVs significantly improves the performance of MAESTRO-X: with 3 UAVs MAESTRO-X delivers the payloads 4.7 \times faster than SCA [10] and 8.6 \times faster than CSCA-ADMM [16]. The gains start to saturate with 2-3 UAVs. In fact, MAESTRO-X approaches the theoretical lower bound to the delay, for large power consumption values: with more power available, UAVs

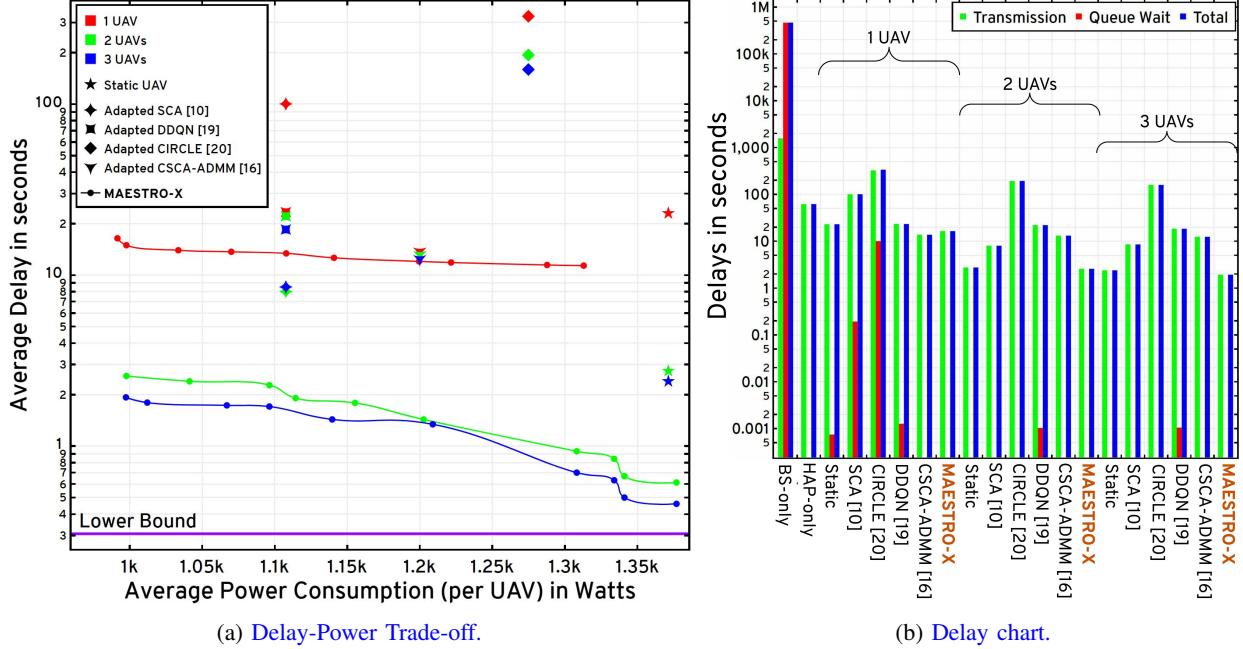


Fig. 5: $L=10$ Mbits, $\Lambda'=0.2$ req/min/UAV: Delay-power trade-off (a) and delay charts (b) for MAESTRO-X, state-of-the-art algorithms, and custom heuristics. In (b), MAESTRO-X is evaluated under $P_{\text{avg}} = 1$ kW.

leverage their mobility to improve pathloss conditions; thanks to spread maximization, multiple UAVs are more likely to be in the vicinity of a request and readily serve it.

In Fig. 5b, we show the contributions of the communication and queue wait times to the overall delay experienced by the GNs, with MAESTRO-X evaluated under a power constraint of 1 kW (less than any other scheme, see Fig. 5a). We note that the BS-only deployment suffers severely due to large communication delays of GNs at the cell edge, causing the queue to become backlogged. The performance is drastically improved by deploying HAPs (HAP-only), thanks to their higher elevation and improved LoS conditions. Yet, the delay performance offered by a HAP-only deployment is poorer than a non-terrestrial deployment involving UAVs: $2.7\times$ slower than a static UAV and $3.8\times$ slower than a UAV optimized with MAESTRO. Across all UAV-assisted implementations, increasing the number of UAVs in the swarm not only lowers the communication delay but also the queue wait times since more GNs can be served simultaneously. Remarkably, MAESTRO-X demonstrates negligible queue wait times even with a single UAV: in this low-traffic regime, requests are served quicker than the rate at which they are generated, thereby bypassing the need for piggybacking and frequency reuse.

To analyze the impact of these mechanisms, in Fig. 6a and Fig. 6b, we study a high congestion regime ($\Lambda'=20$ req/min/UAV). The results depicted in Fig. 6a are qualitatively similar to the low congestion case with some key differences: for all the competitor schemes, we note a performance

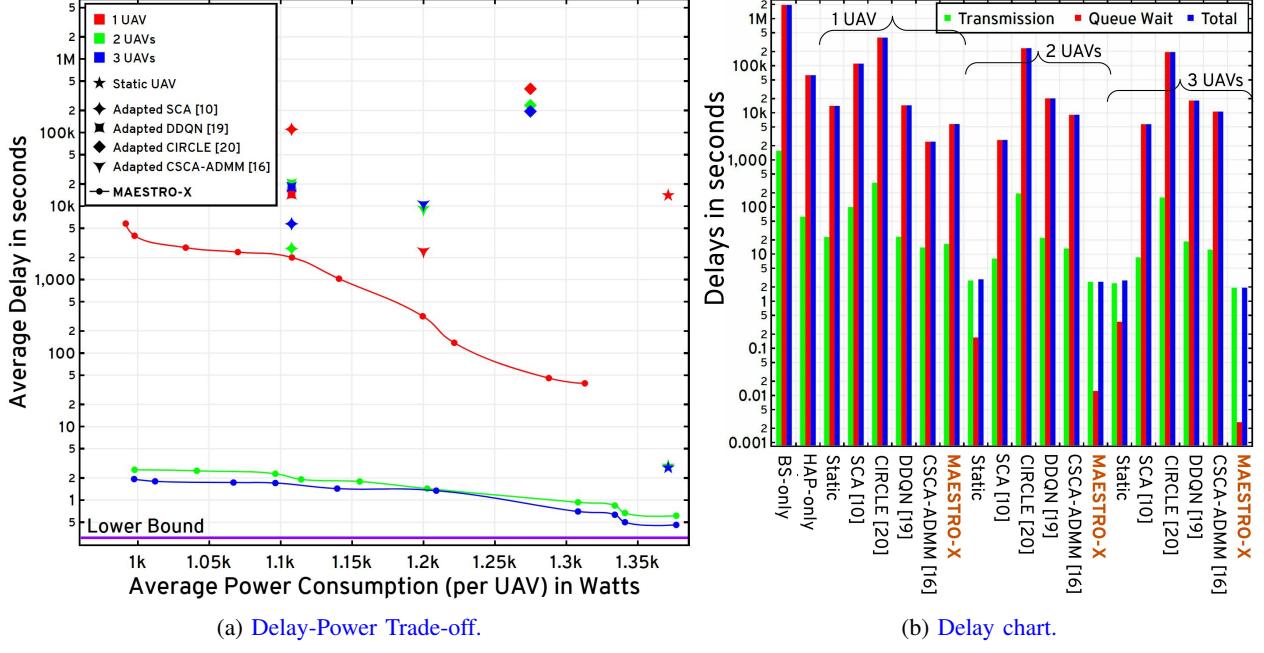


Fig. 6: $L=10$ Mbits, $\Lambda'=20$ req/min/UAV: Delay-power trade-off (a) and delay charts (b) for MAESTRO-X, state-of-the-art algorithms, and custom heuristics. In (b), MAESTRO-X is evaluated under $P_{\text{avg}}=1$ kW.

degradation, due to the large wait times (Fig. 6b); a similar performance degradation is noted for MAESTRO-X with a single UAV. However, remarkably, MAESTRO-X with 2-3 UAVs appears to be unaffected by the higher arrival rate, as also demonstrated by the small queue time. This is attributed to frequency reuse allowing more efficient spectrum use, and to piggybacking allowing simultaneous service of multiple requests by each UAV.

MAESTRO-X, impact of number of channels for large swarms: In Fig. 7, we study the impact of the number of channels (each of 5 MHz) on the average service delay offered by a MAESTRO-X deployment of 10 UAV-relays, in the high congestion regime. Note that the competitors become computationally intractable with more than 5-6 UAVs, whereas the policy replication mechanism of MAESTRO-X offers scalability to large UAV swarms (see Fig. 8). The delay quickly improves by increasing the number of channels, and saturates after 5 channels at 2s delay (consistent with Fig. 6a). This is a remarkable result: for instance, with 4 channels (service delay of ≈ 4 s), if no frequency reuse was allowed, the network could at most service $4[\text{data channels}] \times 15[\text{req/min/data channel}] = 60$ req/min. The ability to serve a much larger rate of $\Lambda = 200$ req/min is attributed to the frequency reuse mechanism.

Policy convergence time: Finally, in Fig. 8, we benchmark MAESTRO-X against SCA from [10] (single-agent, model-based), CSCA-ADMM from [16] (model-based), and DDQN from [19] (model-free), in terms of their policy convergence times, when varying the number of UAVs

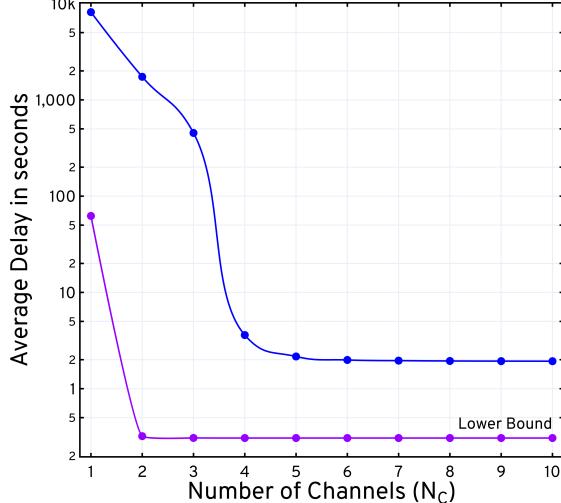


Fig. 7: 10 UAVs, $L=10$ Mbits, $P_{\text{avg}}=1$ kW, $\Lambda=200$ req/min: Average service delay (communication time + queue wait time) vs the number of channels N_C .

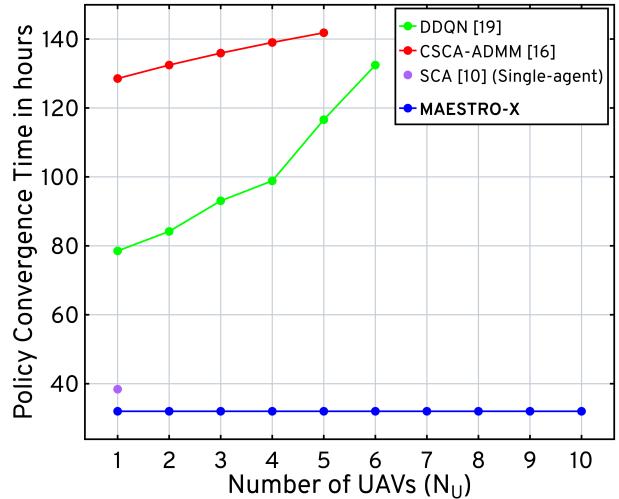


Fig. 8: Policy convergence time (in hours) for MAESTRO-X and the relevant state-of-the-art.

N_U . All implementations are in Python, and are executed on a compute node with 2×64 -core AMD EPYC Milan 7763 CPUs, 16×64 GB DDR4 memory, and $4 \times$ NVIDIA A100 GPUs with 40 GB VRAM each. Remarkably, the convergence time of MAESTRO-X is irrespective of the number of UAVs, whereas it grows quickly for CSCA-ADMM and DDQN. This is due to the policy replication mechanism used by MAESTRO-X: the policy is computed for a single-agent, and then replicated across the swarm, coupled with the supplementary UAV-swarm mechanisms developed in Sec. V. On the other hand, the convergence times of CSCA-ADMM and DDQN grow quickly with the number of UAVs, and become prohibitive when scaled to more than 5 and 6 UAVs, respectively: in fact, it grows linearly for CSCA-ADMM, due to a joint multi-UAV construction involved in its CVXPY-SCS implementation, and exponentially for DDQN, due to combined multi-agent state and action space construction. Remarkably, MAESTRO-X yields a faster convergence time even for a single UAV, thanks to its ability to leverage the multiscale structure of the decision process to achieve a more efficient implementation, in addition to *Tensor*-ized executions exploiting SIMD processing in CUDA-capable GPUs, and distributed workers and thread-pool concurrency in Python (TensorFlow). These benefits in policy convergence coupled with the superior delay-power performance illustrated in Figs. 5 and 6, present MAESTRO-X as an appealing solution for both small and large UAV swarms.

VII. CONCLUSION

In this paper, we propose the MAESTRO-X framework for the decentralized orchestration of rotary-wing UAV-relay swarms in cellular networks, augmenting the coverage and service

capabilities of a terrestrial BS. First, we specialize our system to single-UAV deployments and design the optimal scheduling and trajectory optimization policy under an SMDP formulation. Next, we extend to distributed multi-UAV deployments by employing multi-agent coordination mechanisms, and then replicate this augmented single-UAV policy across the swarm. Numerical evaluations demonstrate that MAESTRO-X delivers significant gains over BS- and HAP-only deployments; furthermore, it exhibits superior performance over static UAV deployments, deep Q-learning schemes, and successive convex approximation strategies.

APPENDIX A: PROOF OF PROP. 1

Since $2(K+1)|g|^2$ has a non-central χ^2 distribution with 2 degrees of freedom and a non-centrality parameter $2K$, we find that $P_{\text{out}}(\Upsilon, \beta, K) = 1 - Q_1(\sqrt{2K}, \sqrt{2(K+1)u(\Upsilon, \beta)})$, where $Q_1(\cdot, \cdot)$ is the standard Marcum Q -function [14]. Hence, $R(\Upsilon, \beta, K) = \Upsilon \cdot Q_1(\sqrt{2K}, \sqrt{2(K+1)u(\Upsilon, \beta)})$. We now maximize it over $\Upsilon \geq 0$. Let $Z \triangleq 2\gamma^{-1}u(\Upsilon, \beta)$ and $\gamma \triangleq \frac{N_0B}{\beta P}$, hence $\Upsilon = B \log_2 \left(1 + \frac{Z}{2}\right) \triangleq f(Z)$. It follows that $\Upsilon^* = f(Z^*)$, where Z^* maximizes over $Z \geq 0$ the function

$$h(Z) \triangleq \ln R(f(Z), \beta, K) = \ln f(Z) + \ln Q_1(\sqrt{2K}, \sqrt{\gamma(K+1)Z}). \quad (22)$$

Note that $Q_1(a, \sqrt{bZ})$ is log-concave in $Z \geq 0$ for $a, b > 0$ (see [40]), and second derivative of $\ln f(Z)$ satisfies $(\ln f(Z))'' = \frac{f''(Z)}{f(Z)} - \frac{(f'(Z))^2}{(f(Z))^2} \leq 0, \forall Z \geq 0$, so that $h(Z)$ is concave in $Z \geq 0$. Since $\lim_{Z \rightarrow 0^+} h(Z) = -\infty$ and $\lim_{Z \rightarrow \infty} h(Z) = -\infty$, there exists a unique $Z^* \in (0, \infty)$ (hence $\Upsilon^* = f(Z^*)$) such that $h'(Z^*) = 0$, solvable with the bisection method, with $h'(Z)$ given by

$$h'(Z) = \frac{f'(Z)}{f(Z)} + \frac{\sqrt{\gamma(K+1)}}{2\sqrt{Z}} \frac{\partial Q_1(\sqrt{2K}, b)/\partial b|_{b=\sqrt{\gamma(K+1)Z}}}{Q_1(\sqrt{2K}, \sqrt{\gamma(K+1)Z})},$$

yielding (1) after solving for f' and the partial derivative of Q_1 .

APPENDIX B: PROOF OF PROP. 2

Let $\bar{W}_\mu \triangleq \bar{W}_\mu^{(s)} + \bar{W}_\mu^{(bs)}$. If $\xi_u = 1$, then additional requests received during the UAV relay phase are served directly by the BS, with delay $\frac{L}{R_{GB}(r)}$ for a GN in position (r, θ) . Thus, the expected average communication delay to serve these additional requests is $\mathbb{E}[\Delta_{u,i}^{(bs)}] = \bar{D}_{BS}$, yielding $\bar{W}_\mu = \bar{W}_\mu^{(s)} + \bar{D}_{BS}(\bar{N}_\mu - 1)$ and $\bar{D}_\mu = \frac{\bar{W}_\mu}{N_\mu} = \frac{\bar{W}_\mu^{(s)}}{N_\mu} + \left(1 - \frac{1}{N_\mu}\right) \bar{D}_{BS}$. Let μ be any policy (including the optimal one) that satisfies $\bar{D}_\mu \leq \bar{D}_{BS}$: under such policy, since $\bar{N}_\mu \geq 1$, the expression above implies that $\bar{W}_\mu^{(s)} \leq \bar{D}_\mu \leq \bar{D}_{BS}$. Moreover, since $\mathbb{E}[N_u | \Delta_u^{(s)}] = \Delta_u^{(s)} \Lambda'$ and $\xi_u \leq 1$, it follows that $\bar{N}_\mu \leq 1 + \Lambda' \bar{W}_\mu^{(s)}$ with equality if the UAV always serves requests. This implies (8).

APPENDIX C: PROOF OF PROP. 3

Let $\pi_{\text{wait}}=1-\pi_{\text{comm}}$ be the SMDP steady-state probability of the UAV being in the waiting state. Since the probability of remaining in the waiting state (no request is received) in one SMDP step is $p_{ww}=e^{-\Lambda'\Delta_0}$ and that of moving from a communication state to a waiting state is $p_{cw}=1$, π_{comm} and π_{wait} are solutions of the stationary equation $\pi_{\text{wait}} = \pi_{\text{wait}}p_{ww} + \pi_{\text{comm}}p_{cw} = e^{-\Lambda'\Delta_0}\pi_{\text{wait}} + \pi_{\text{comm}}$. Solving it with $\pi_{\text{wait}}+\pi_{\text{comm}}=1$ yields the expression of π_{comm} in Prop. 3.

REFERENCES

- [1] B. Keshavamurthy and N. Michelusi, “Multiscale Adaptive Scheduling and Path-Planning for Power-Constrained UAV-Relays via SMDPs,” 2022. [Online]. Available: <https://arxiv.org/abs/2209.07655>
- [2] B. Keshavamurthy, “MAESTRO-X: Multiscale Adaptive Energy-conscious Scheduling and TRajecoty Optimization (eXtended),” April 2022. [Online]. Available: <https://github.com/bharathkeshavamurthy/MAESTRO-X.git>
- [3] A. Fotouhi, H. Qiang *et al.*, “Survey on UAV Cellular Communications: Practical Aspects, Standardization Advancements, Regulation, and Security Challenges,” *IEEE Communications Surveys Tutorials*, pp. 1–1, 2019.
- [4] M. Mozaffari, W. Saad *et al.*, “A Tutorial on UAVs for Wireless Networks: Applications, Challenges, and Open Problems,” *IEEE Communications Surveys Tutorials*, pp. 1–1, 2019.
- [5] B. Keshavamurthy and N. Michelusi, “Learning-Based Spectrum Sensing and Access in Cognitive Radios via Approximate POMDPs,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 8, no. 2, pp. 514–528, 2022.
- [6] D. Alvear, “Inside Verizon | Up to Speed | Drones, robots and the power of 5G,” July 2021.
- [7] S. Guillette, “Verizon News Archives | Robots, drones and sensors are changing the way we farm,” March 2019.
- [8] Y. Zeng, R. Zhang *et al.*, “Wireless communications with unmanned aerial vehicles: opportunities and challenges,” *IEEE Communications Magazine*, vol. 54, no. 5, pp. 36–42, May 2016.
- [9] Q. Wu, L. Liu *et al.*, “Fundamental Trade-offs in Communication and Trajectory Design for UAV-Enabled Wireless Network,” *IEEE Wireless Communications*, vol. 26, pp. 36–44, 02 2019.
- [10] Y. Zeng, J. Xu *et al.*, “Energy Minimization for Wireless Communication With Rotary-Wing UAV,” *IEEE Transactions on Wireless Communications*, vol. 18, no. 4, pp. 2329–2345, April 2019.
- [11] M. A. Abd-Elmagid and H. S. Dhillon, “Average Peak Age-of-Information Minimization in UAV-Assisted IoT Networks,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 2003–2008, 2019.
- [12] X. Hu, K.-K. Wong *et al.*, “UAV-Assisted Relaying and Edge Computing: Scheduling and Trajectory Optimization,” *IEEE Transactions on Wireless Communications*, vol. 18, no. 10, pp. 4738–4752, 2019.
- [13] J. Chen and D. Gesbert, “Optimal positioning of flying relays for wireless networks: A LOS map approach,” in *2017 IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–6.
- [14] C. You and R. Zhang, “3D Trajectory Optimization in Rician Fading for UAV-Enabled Data Harvesting,” *IEEE Transactions on Wireless Communications*, vol. 18, no. 6, pp. 3192–3207, 2019.
- [15] R. K. Patra and P. Muthuchidambaranathan, “Optimisation of Spectrum and Energy Efficiency in UAV-Enabled Mobile Relaying Using Bisection and PSO Method,” in *3rd Int. Conference for Convergence in Technology (I2CT)*, 2018, pp. 1–7.
- [16] Q. Hu, Y. Cai *et al.*, “Low-Complexity Joint Resource Allocation and Trajectory Design for UAV-Aided Relay Networks With the Segmented Ray-Tracing Channel Model,” *IEEE Transactions on Wireless Communications*, vol. 19, no. 9, 2020.
- [17] Q. Wu, Y. Zeng *et al.*, “Joint Trajectory and Communication Design for Multi-UAV Enabled Wireless Networks,” *IEEE Transactions on Wireless Communications*, vol. 17, no. 3, pp. 2109–2121, 2018.

- [18] M. Mozaffari, W. Saad *et al.*, “Efficient Deployment of Multiple Unmanned Aerial Vehicles for Optimal Wireless Coverage,” *IEEE Communications Letters*, vol. 20, no. 8, pp. 1647–1650, 2016.
- [19] H. Bayerlein, M. Theile *et al.*, “Multi-UAV Path Planning for Wireless Data Harvesting With Deep Reinforcement Learning,” *IEEE Open Journal of the Communications Society*, vol. 2, p. 1171–1187, 2021.
- [20] L. Wang, K. Wang *et al.*, “Multi-Agent Deep Reinforcement Learning-Based Trajectory Planning for Multi-UAV Assisted Mobile Edge Computing,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, no. 1, 2021.
- [21] A. M. Koushik, F. Hu *et al.*, “Deep Q-Learning-Based Node Positioning for Throughput-Optimal Communications in Dynamic UAV Swarm Network,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 3, 2019.
- [22] Q. Zhang, M. Mozaffari *et al.*, “Machine Learning for Predictive On-Demand Deployment of UAVs for Wireless Communications,” in *2018 IEEE Global Communications Conference (GLOBECOM)*, 2018, pp. 1–6.
- [23] A. E. A. A. Abdulla, Z. M. Fadlullah *et al.*, “Toward Fair Maximization of Energy Efficiency in Multiple UAS-Aided Networks: A Game-Theoretic Methodology,” *IEEE Transactions on Wireless Communications*, vol. 14, no. 1, Jan 2015.
- [24] Y. Li and L. Cai, “UAV-Assisted Dynamic Coverage in a Heterogeneous Cellular System,” *IEEE Network*, vol. 31, no. 4, pp. 56–61, July 2017.
- [25] D. P. Bertsekas, *Dynamic Programming and Optimal Control, Vol. II*, 4th ed. Athena Scientific, 2007.
- [26] R. Cheng and Y. Jin, “A Competitive Swarm Optimizer for Large Scale Optimization,” *IEEE Transactions on Cybernetics*, vol. 45, no. 2, pp. 191–204, 2015.
- [27] J. Hu, H. Zhang *et al.*, “Reinforcement Learning for Decentralized Trajectory Design in Cellular UAV Networks With Sense-and-Send Protocol,” *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6177–6189, 2019.
- [28] C. Zhou, H. He *et al.*, “Deep RL-based Trajectory Planning for AoI Minimization in UAV-assisted IoT,” in *2019 11th International Conference on Wireless Communications and Signal Processing (WCSP)*, 2019, pp. 1–6.
- [29] M. Clerc, *Particle Swarm Optimization*, ser. ISTE. London ; Newport Beach: ISTE, 2006.
- [30] H. Shakhatreh, A. Khreichah *et al.*, “Efficient 3D placement of a UAV using particle swarm optimization,” in *2017 8th International Conference on Information and Communication Systems (ICICS)*, 2017, pp. 258–263.
- [31] Z. Yuheng, Z. Liyan *et al.*, “3-D Deployment Optimization of UAVs Based on Particle Swarm Algorithm,” in *2019 IEEE 19th International Conference on Communication Technology (ICCT)*, 2019, pp. 954–957.
- [32] A. Al-Hourani, S. Kandeepan *et al.*, “Optimal LAP Altitude for Maximum Coverage,” *IEEE Wireless Communications Letters*, vol. 3, no. 6, pp. 569–572, 2014.
- [33] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge: Cambridge University Press, 2005.
- [34] R. Essaadali and A. Kouki, “A new simple Unmanned Aerial Vehicle doppler effect RF reducing technique,” in *MILCOM 2016 - 2016 IEEE Military Communications Conference*, 2016, pp. 1179–1183.
- [35] J. D. Little and S. C. Graves, “Little’s Law,” in *Building Intuition*. Springer, 2008, pp. 81–100.
- [36] S. Boyd, L. Xiao *et al.*, “Subgradient methods,” *Lecture notes, Stanford University, Autumn Quarter*, pp. 2004–2005, 2003.
- [37] Y. Zeng and R. Zhang, “Energy-Efficient UAV Communication With Trajectory Optimization,” *IEEE Transactions on Wireless Communications*, vol. 16, no. 6, pp. 3747–3760, June 2017.
- [38] P. Yang, K. Tang *et al.*, “Turning High-Dimensional Optimization Into Computationally Expensive Optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 143–156, 2018.
- [39] A. Al-Hourani, S. Kandeepan *et al.*, “Optimal LAP Altitude for Maximum Coverage,” *IEEE Wireless Communications Letters*, vol. 3, no. 6, pp. 569–572, 2014.
- [40] Y. Sun and S. Zhou, “Tight bounds of the generalized marcum q-function based on log-concavity,” in *IEEE GLOBECOM 2008 - 2008 IEEE Global Telecommunications Conference*, 2008, pp. 1–5.

RESPONSES TO THE EDITOR AND THE REVIEWERS

We would like to thank the Editor and the Reviewers for their careful assessment of the manuscript and for their comments, which we found very useful to improve the quality of the manuscript. We have addressed all the comments in the revised version to the best of our abilities, and you will find enumerated responses to your comments below. Your comments are in italics, our response follows after "**R:**".

To facilitate the second round of revisions, we highlighted the major changes using **blue font** in the manuscript. Although the manuscript has been thoroughly revised for clarity, consistence of notation, etc., minor typographical and sentence-structural changes have not been highlighted.

Regards,

Bharath Keshavamurthy, Matthew Bliss, and Dr. Nicolò Michelusi

EDITOR

We thank the Editor for the timely completion of the review process. We have addressed all the Reviewers' comments to the best of our abilities, carrying out a major revision of the system model, along with a number of editorial changes and additional explanations. We believe that this version of the paper is much improved. As per the Reviewers' suggestions,

- We have highlighted the contributions of our work with respect to the existing state-of-the-art more clearly in the introduction (Sec. I) by adding additional modeling features for comparison to our formulation in Table I. Likewise, we have added a contributions section to the introduction (Sec. I, page 5) that explicitly highlights the technical challenges solved by our novel MAESTRO-X solution framework.
- We have improved the readability of the manuscript by reorganizing the analytical results into propositions and moving their proofs to appendices.
- We have conducted more extensive numerical evaluations, and have increased the number of use cases and parameters tested. In addition, we have improved the quality of figures in the numerical evaluations, Sec. VI.
- We have improved the system spectrum utilization efficiency by enabling a frequency reuse mechanism (see description in Sec. V), wherein UAVs outside of the interference region of other UAVs are able to share data channels simultaneously to serve their respective GN requests.
- We have removed the restrictive assumption in our multi-UAV formulation that each UAV may only serve up to one GN uplink request at a time. In this revised manuscript, we have devised a piggybacking mechanism (see Sec. V), incorporated into the consensus-driven conflict resolution process, wherein UAVs can serve new GN requests while simultaneously serving past requests and following their collated optimal service trajectories.

A detailed point-to-point response to the reviewers follows.

REVIEWER 1

- 1) **Comment:** *The contribution of this work is not very clear. It seems like many issues, such as dynamic data traffic from miscellaneous sets of randomly distributed ground users, and a feasible trajectory design solution without unrealistic assumptions on UAV mobility, are considered, but what are the technical challenges of the integrated study and what is new in the solution? For example, the value iteration method adopted in Algorithm 1 is well-known and the novelty is not clear. It would be better to clarify them.*

R: We thank the reviewer for the valuable assessment and have addressed the lack of clarity in explaining the technical contributions of this work. We would like to remark that, in our work, we do not blindly apply a well-known algorithm such as value iteration to our framework. Such a blind approach would entail extremely large action (UAV trajectories) and state spaces (UAV and GN positions), leading to prohibitively large policy convergence times, a problem commonly seen in the relevant state-of-the-art (e.g., [16], [19]). Instead, we carefully leverage the structure of the problem to optimize the decision-making process. In particular, our careful choice of states and actions leads to the two-scale decomposition of the decision-making process into outer and inner actions. For instance, in communication states, thanks to this decomposition, the outer action optimized via value iteration need only specify a target end radius during the communication phase (rather than an entire trajectory); given such outer action, the inner action (the actual trajectory of the UAV) can be optimized efficiently (via our newly proposed HCSO) by greedy minimization of a Lagrangian cost metric, capturing the trade-off between delay and energy consumption.

Note also that the SMDP formulation is non-trivial in our setup: it required us to devise an alternative optimization problem (see discussion on page 11, now also validated numerically in Table III), along with a careful definition of states, actions, and when states are sampled in time to obtain the Markovian dynamics of SMDPs (see Sec. III).

Regarding the utility of our novel HCSO algorithm (Alg. 3), we note that a direct application of CSO alone would suffer from a complexity-accuracy dilemma [26], as also noted in the revised manuscript, page 20:

"[...] high-resolution trajectories are slow to converge, while low-resolution ones give rise to poor solutions that fail to capture fine-grained variations in the trajectory way-points and velocities. To overcome this limitation, we embed CSO within a hierarchical wrapper:

starting from a low-resolution trajectory optimized via CSO, after each CSO iteration, the resulting trajectory is interpolated to form a higher-resolution reference trajectory. The swarm size is reduced, and a new set of particles are generated randomly in the neighborhood of the reference trajectory, in order to preserve the quality of previous lower-resolution trajectories. This process repeats until the desired resolution (number of way-points and velocities) is achieved."

In addition to issues such as dynamic data traffic and high-resolution trajectory design, the primary technical challenge addressed in our work is to create a framework for the *scalable* orchestration of UAV-relay swarms. While state-of-the-art model-free approaches in the research literature, often applying deep reinforcement learning (see [19]–[22]), consider unknown system dynamics in their formulations to solve for the optimal trajectory design and/or scheduling solution, they fail to efficiently capture problem structure, which results in slow policy convergence times and ultimately limits the ability to scale the policy implementation to multi-UAV formulations.

In contrast, our approach exploits the multiscale structure of a model-based single-agent SMDP formulation, which requires the careful definition of state and actions in such a way so as to ensure the proper definition of a Markovian process. Consequently, this enables the application of the value iteration algorithm. Then, to facilitate an efficient extension to distributed multi-UAV deployments with decoupled scheduling decisions, we replicate the single-agent policy across the swarm, combined with multi-agent mechanisms enacted over a decentralized command-and-control network (e.g., spread maximization, consensus-driven conflict resolution, as well as the newly-added adaptive frequency reuse and piggybacking mechanisms, described in Sec. V) in order to enable a scalable multi-UAV design.

In this document addressing the reviewers' comments, Fig. 9 depicts our solution approach (not included in the manuscript due to space constraints), highlighting our main contributions as "novelties". In the manuscript, we have revised the introduction (Sec. I) to include a contributions section (see page 5) to explicitly highlight these technical contributions: these revisions are highlighted in Sec. I on pages 2, 3 and 5.

Finally, Table I (Sec. I on page 3) details the necessary set of features essential for efficient deployments of UAV-relay swarms for non-terrestrial augmentations of conventional radio

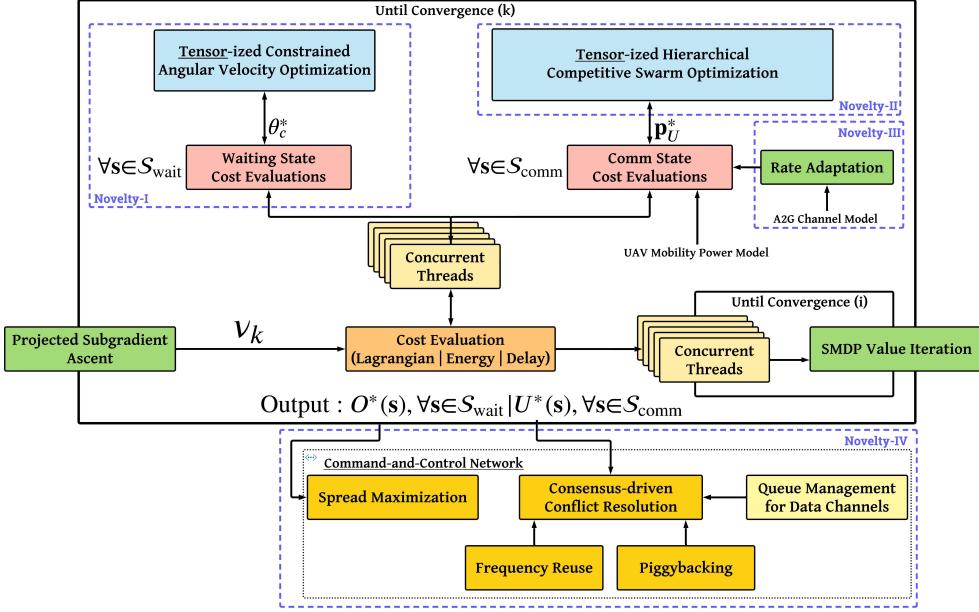


Fig. 9: An illustration depicting the framework proposed in this paper, highlighting the relevant contributions. access networks: these constitute an enumerated list of the technical challenges in our integrated study, along with the other aspects of our research discussed above.

- 2) **Comment:** *Also, it would be better if the complexity of the proposed solution can be discussed, as some proposed algorithms (e.g., Algorithm 1 and Algorithm 3) seem to have very high complexity. How does it scale with the number of UAVs?*

R: We thank the reviewer for the valuable assessment. In addressing Alg. 1 (value iteration) and Alg. 3 (HCSO) in relation to the computational complexity, we present a worst-case complexity analysis in Table IV for reference (the table is not included in the manuscript due to page constraints). The parameters listed in the table are explained as follows:

- Common parameters: N_U (number of UAVs), N_G (number of GNs);
- DDQN: T (total number of mission time-steps after discretizing simulation time period), M_g^2 (number of grid spaces after discretizing the deployment region into a rectangular grid);
- CSCA-ADMM: I_{\max} (maximum number of iterations), N_T (number of time-steps upon discretizing the mission duration), l_0 (initialized interval length for bisection search to obtain Lagrangian multipliers), ϵ (minimum required cost convergence accuracy);
- MAESTRO-X: K_R (number of discretized radii values), K_A (number of discretized angular values), K_V (number of discretized radial velocity levels), N (initial swarm

Framework	Worst-Case Complexity
DDQN [19]	$\mathcal{O}\left(T(N_U)^{M_g^2} N_G 6^{N_U}\right)$
CSCA-ADMM [16]	$\mathcal{O}\left(N_U I_{\max}^2 N_T N_G \left[1 + \log_2\left(\frac{\ell_0}{\epsilon}\right)\right]\right)$
MAESTRO-X	$\mathcal{O}\left[K_R \left(K_V + K_R^2 K_A\right) N^2 K_{\max} M_{\max}\right]$

TABLE IV: Complexity Comparisons across the relevant state-of-the-art.

size in HCSO), K_{\max} (maximum number of cost function evaluations in CSO), M_{\max} (maximum number of trajectory segments in HCSO).

All framework complexities shown in Table IV correspond to the full multi-UAV implementation with N_U (number of UAVs in the swarm)—specifically, we compare the Big-O algorithmic complexity of DDQN (Double Deep-Q Network) [19] and CSCA-ADMM (Constrained SCA with Alternating Direction Method of Multipliers) [16] to that of our MAESTRO-X framework.

Studying the dependence of worst-case complexities on the number of UAVs N_U , we observe that for both the DDQN [19] and CSCA-ADMM [16] frameworks, the worst-case complexity scales with the number of UAVs N_U , and for DDQN in particular, it scales exponentially with N_U due to the fact that a large joint state and action space is considered. In contrast, the number of UAVs N_U has no influence on the worst-case complexity of MAESTRO-X. This is due to the fact that the MAESTRO-X formulation is built on the principle of policy replication, performed for a single-UAV scenario, and is then augmented in the multi-UAV setting by using a decentralized command-and-control network to implement the consensus-driven conflict resolution, spread maximization, frequency reuse, and piggybacking mechanisms. To further validate this behavior, Fig. 8 in the manuscript depicts the policy convergence time (in hours) for these algorithms as a function of the number of UAVs; these results align with the worst-case complexity provided in Table IV, and it is clear that the convergence time of MAESTRO-X is uninfluenced by N_U , just as the worst-case complexity analysis predicts.

In addressing optimality guarantees of the HCSO algorithm, we note that the constrained problem **(P.0)** (see page 19) that is solved with HCSO is non-convex in both the objective and constraints. In the absence of further problem structure, iterative optimization methods, including HCSO (which relies internally on CSO), applied to this problem cannot guarantee convergence to a global optimum. However, as depicted below, these algorithms do possess convergence guarantees to local optima:

- CSO: Theoretical convergence guarantees to a local optimum and computational complexity benchmarks are provided in [26] via stability analysis for dynamical systems. Therein, a straightforward analysis of the eigenvalues of the dynamical system's state matrix show that a sufficient condition for convergence to an equilibrium point is met when the scale factor ϕ (the hyper-parameter controlling the influence that the mean particle position in a given iteration has on the particle velocity update equation) satisfies $\phi \geq 0$.
- HCSO: Relies on CSO internally with multiple CSO calls (each guaranteed to converge to a local optimum, based on [26] mentioned in the previous item) iteratively executed until a trajectory of a certain pre-defined resolution is obtained, hence we conclude that HCSO inherits the convergence properties of CSO.

We have addressed this comment about complexity in the manuscript as quoted below.

"In practice, the integrals in lines 16 and 17, and the continuous state/action spaces are discretized (see MAESTRO-X [2]), leading to an overall complexity of each value iteration update (lines 5-18) of order $\mathcal{O}(K_R \cdot (K_V + K_R^2 \cdot K_A))$, where K_R is the number of discretized radii levels (r_U and r values), K_A is the number of angular levels (ψ and ψ'), and K_V is the number of discretized radial velocities (v_r). Upon convergence (typically, value iteration converges within $\mathcal{O}(\log(1/\delta))$ iterations to achieve a target accuracy δ [25, Sec. V]), line 21 estimates the values of the average cost-per-stage and excess energy-per-stage metrics." (Sec. III on page 17)

In addition, the numerical convergence time comparison can be found in Fig. 8 and related discussion (Sec. VI on page 29). In particular,

"On the other hand, the convergence times of CSCA-ADMM and DDQN grow quickly with the number of UAVs, and become prohibitive when scaled to more than 5 and 6 UAVs, respectively: in fact, it grows linearly for CSCA-ADMM, due to a joint multi-UAV construction involved in its CVXPY-SCS implementation, and exponentially for DDQN, due to combined multi-agent state and action space construction."

- 3) **Comment:** *To enhance readability, it would be better if the main results can be summarized as propositions/theorems and their proofs can be put into appendices.*

R: Following the Reviewer's suggestion, we have summarized our analytical results as

propositions, with their proofs placed into appendices. Specifically, the solution of the throughput-maximizing rate used for rate adaptation is now stated in Proposition 1 (Sec. II on page 8); its proof is located in Appendix A (on page 30). Next, we reformulate the derivation of upper and lower bounds on \bar{D}_μ in Proposition 2 (Sec. III on page 12); the corresponding proof is located in Appendix B (on page 30). Finally, we reformulate the derivation of the steady-state probability π_{comm} that the UAV is in the SMDP communication state in Proposition 3 (Sec. III on page 15), proved in Appendix C (on page 31).

- 4) **Comment:** *It is claimed that the proposed scheme is scalable. However, only up to 3 UAVs are considered in some simulations (e.g., Fig. 6). It would be better if more UAVs can be considered in these figures.*

R: We thank the reviewer for pointing out this issue. We structure our response as follows. First, we comment on the scalability of MAESTRO-X with the number of UAVs; second, we motivate the limitation to 3 UAVs in our results; third, we present a new numerical evaluation to 10 UAVs, included in the revised manuscript.

1) As shown in Table IV, the worst-case complexity of MAESTRO-X is uninfluenced by the number N_U of UAVs in the swarm, thanks to the policy replication mechanism employed, and augmented with multi-UAV coordination mechanisms over a decentralized command-and-control network. We refer the reviewer to our response to Comment 2 and to Table IV for a discussion on the scalability of MAESTRO-X and the state-of-the-art algorithms, confirmed by Fig. 8. In particular, note that the state-of-the-art approaches (CSCA-ADMM [16] and DDQN [19]) become prohibitively expensive as the number of UAVs in the swarm increases.

2) While analyzing the delay-power performance of MAESTRO-X in Figs. 5a and 6a, we observe that the performance saturates as the number of UAVs increases to 3, and approaches a (newly added) theoretical lower bound to the service delay (*Lower Bound* in the figure; it is described in the highlighted text on page 26). A similar saturation effect is visible for the other schemes as well. This indicates that increasing the number of UAVs in the swarm further is expected to have a negligible impact on the delay performance (in any case, it cannot be any lower than the newly added lower bound). Note also that, as pointed out in the previous point and shown in Fig. 8, some state-of-the-art schemes become prohibitively

expensive even with a few UAVs. For these reasons, we decided to limit these evaluations to 3 UAVs to allow a fair comparison with the state-of-the-art schemes.

- 3) Following the Reviewer's suggestion, we have included a new evaluation in the revised manuscript, showing the impact of the number of channels with MAESTRO-X operated on a swarm of 10 UAVs (see Fig. 7 and related discussion on page 29), which demonstrates the effectiveness of the newly added frequency reuse mechanism.
- 5) **Comment:** *The assumed orthogonal channels may lead to low spectrum efficiency and the bandwidth per link may quickly drop when the number of UAVs increases. It would be better if the authors can provide some discussions on this aspect.*

R: We thank the reviewer for the valuable assessment. In our original manuscript, each UAV and the BS had dedicated channels resulting in low spectrum utilization efficiency. In the revised manuscript, the total application-wide bandwidth (W) is discretized into N_C orthogonal data channels, independent of the number of UAVs and GNs in the deployment. When a new GN request arises in the cell, the server nodes (UAVs and BS) coordinate over the control channels (band-edges of the spectrum) for data channel assignment, cost-of-service computation, and a subsequent consensus on the best server node to handle the new request (Sec. II on page 6). Furthermore, in our original manuscript, at a UAV, the same dedicated channel was used for both decode and forward phases of its service operations. In the revised manuscript, there are no such restrictions; instead, modeling the channel access behavior as queues, we conduct dynamic channel assignments for each phase of the UAV's operations. Additionally, in our original manuscript, a UAV was restricted to serve only one GN at a time; we have upgraded this setup to allow the UAVs to simultaneously serve multiple GNs via a new piggybacking mechanism (Sec. V on page 23). Also, in order to further enhance the spectrum utilization efficiency across our network, we have incorporated an adaptive frequency reuse mechanism that allows data channels to be reused by different Tx-Rx pairs, as long as an SINR degradation condition is met (equivalent to a minimum distance condition; Sec V on page 22).

Accordingly, we created additional numerical evaluation scenarios to test the effectiveness of these newly added mechanisms:

- We realized that, in our original manuscript, we only simulated $L=1$ Mb data payloads

under a low congestion regime ($\Lambda'=1$ req/min), in which spectrum efficiency was not an issue. Enabled by these additional piggybacking and frequency reuse mechanisms, in the revised manuscript we evaluated numerically the performance with a larger data payload of $L=10$ Mbits under both a low congestion regime (Figs. 5a and 5b in Sec. VI on page 27) as well as a high congestion regime (Figs. 6a and 6b in Sec. VI on page 28). This new regime demonstrates the effectiveness of these mechanisms in handling incoming data traffic and getting it delivered with small queuing delay (see discussion accompanying Figs. 6a and 6b).

- Finally, in the revised manuscript, we also evaluate the impact of the number of channels on the average delay performance of 10 UAVs for $L=10$ Mbits under a high congestion regime of $\Lambda=200$ req/min (see Fig. 7 in Sec. VI on page 29 and accompanying discussion). This figure demonstrates that the system achieves good delay performance with only 4 data channels, which are effectively shared among the BS and the 10 UAVs.

In addition to the new numerical evaluations, this comment has been addressed throughout the manuscript as follows.

"Multi-user communication is enabled via OFDMA over a spectrum of bandwidth W , discretized into N_C orthogonal data channels (possibly, obtained by grouping multiple subcarriers together), each with bandwidth $B \triangleq \frac{W}{N_C}$." (Sec II on page 6)

"A fully-connected mesh network overlaying the BS and UAVs enables command-and-control using the band-edges of the allocated spectrum as control channels. Since control packets constitute short frames relative to the large GN-generated data payloads (communicated over data channels), the control operation latencies are neglected. [...] In Sec. V, we also discuss a *frequency reuse* mechanism to improve spectrum utilization efficiency, and a *piggybacking* mechanism allowing the scheduled UAV to serve multiple requests simultaneously." (Sec II on page 6)

Piggybacking: To facilitate simultaneous multi-user service at the UAVs, we incorporate a piggybacking mechanism (in the cost-of-service computation of the consensus-driven conflict resolution process), wherein a UAV follows a collated trajectory to accommodate new GN uplink requests while serving previous requests. [...]" (Sec. V on page 23)

Frequency Reuse: To improve the spectrum utilization efficiency, we propose a

frequency reuse mechanism, allowing multiple serving nodes (the BS and UAVs) to share the same data channel simultaneously when serving their respective GN requests. [...] (Sec. V on page 22)

REVIEWER 2

- 1) **Comment:** *The Algorithms 1 and 2 utilize swarm optimization which is a heuristic and the authors do not provide any optimality guarantees.*

R: We thank the reviewer for the valuable assessment. In addressing optimality guarantees of the HCSO algorithm, we note that the constrained problem (**P.0**) (see page 19) that is solved with HCSO is non-convex in both the objective and constraints. In the absence of further problem structure, iterative optimization methods, including HCSO (which relies internally on CSO), applied to this problem cannot guarantee convergence to a global optimum. However, as depicted below, these algorithms do possess convergence guarantees to local optima:

- CSO: Theoretical convergence guarantees to a local optimum are provided in [26] via stability analysis for dynamical systems. Therein, a straightforward analysis of the eigenvalues of the dynamical system's state matrix show that a sufficient condition for convergence to an equilibrium point is met when the scale factor ϕ (the hyper-parameter controlling the influence that the mean particle position in a given iteration has on the particle velocity update equation) satisfies $\phi \geq 0$.
- HCSO: Relies on CSO internally with multiple CSO calls (each guaranteed to converge to a local optimum, based on [26] mentioned in the previous item) iteratively executed until a trajectory of a certain pre-defined resolution is obtained, hence we conclude that convergence to a local optimum is guaranteed for HCSO as well.

- 2) **Comment:** *The simulations in this work are not enough. More use cases and parameters should be shown. Also, more comparisons with the previous works should be given. The quality of the figures should be improved as well.*

R: Following the Reviewer's concerns, we have evaluated additional use cases in the revised manuscript and added new state-of-the-art evaluations: CIRCLE [20], which involves heuristics to cluster GNs for simultaneous multi-user service as well as to design the associated optimal radius value for a UAV's circular trajectory, and a *HAP-only* setting, in which GNs transmit directly to a High Altitude Platform (a popular non-terrestrial augmentation approach in next-generation wireless networks). Note that our revised numerical results now include comparisons with 4 state-of-the-art schemes and 3 heuristic schemes. Due to space

constraints, we were not able to include all of the new results in the revised manuscript; however, we have provided these additional results in this response, as detailed next. We have also made an effort to improve the readability of the new figures. We hope that our modifications are satisfactory.

Regarding the new numerical evaluations, the following additions have been made with respect to the original manuscript:

- a) the revised manuscript now includes an evaluation of both low-congestion and high-congestion regimes (see Figs. 5a, 5b and Figs. 6a, 6b), whereas the revised manuscript only included a low-congestion evaluation;
- b) We have included a numerical evaluation in Table III, to demonstrate the validity of the alternative optimization problem developed in Sec. III;
- c) In Fig. 7, we have included a numerical evaluation with a swarm of 10 UAVs, depicting the impact of the number of channels, and the effectiveness of two newly added mechanisms: frequency reuse and piggybacking, detailed in Sec. VI.
- d) We have also revised Fig. 4b, which now includes a visualization of additional trajectories in the communication phase, optimized via HCSO.

We refer to the appropriate subsections in Sec. VI for a discussion of these results. We have also performed numerical evaluations in a high-payload case of $L=100$ Mbits. These results are qualitatively similar to the case $L=10$ Mbits, as discussed next. For this reason and due to space constraints, this large payload scenario has not been included in the revised manuscript.

Large payload evaluations (not included in the manuscript, due to space constraints): Fig. 4a in the revised manuscript (replicated in Fig. 10a in this response document) depicts the optimal waiting policy for $L=10$ Mbits, $\Lambda'=0.2$ req/min/UAV, and $P_{avg}=1.2$ kW. We have commented on this figure in Sec. VI on page 25. In this response, we also show Fig. 10b, depicting the optimal waiting policy for $L=100$ Mbits, $\Lambda'=0.033$ req/min/UAV, and $P_{avg}=1.2$ kW: we notice a very similar behavior as the smaller payload case, demonstrating that the waiting policy is not much sensitive with respect to variations in payload size.

Fig. 11a in this response depicts an evaluation of the delay-power trade-off for $L=100$ Mbits data payloads ($\Lambda'=0.033$ req/min/UAV): here, we evaluate the performance of MAESTRO-X against deployment heuristics (BS, HAP, static-UAV) and against relevant schemes in

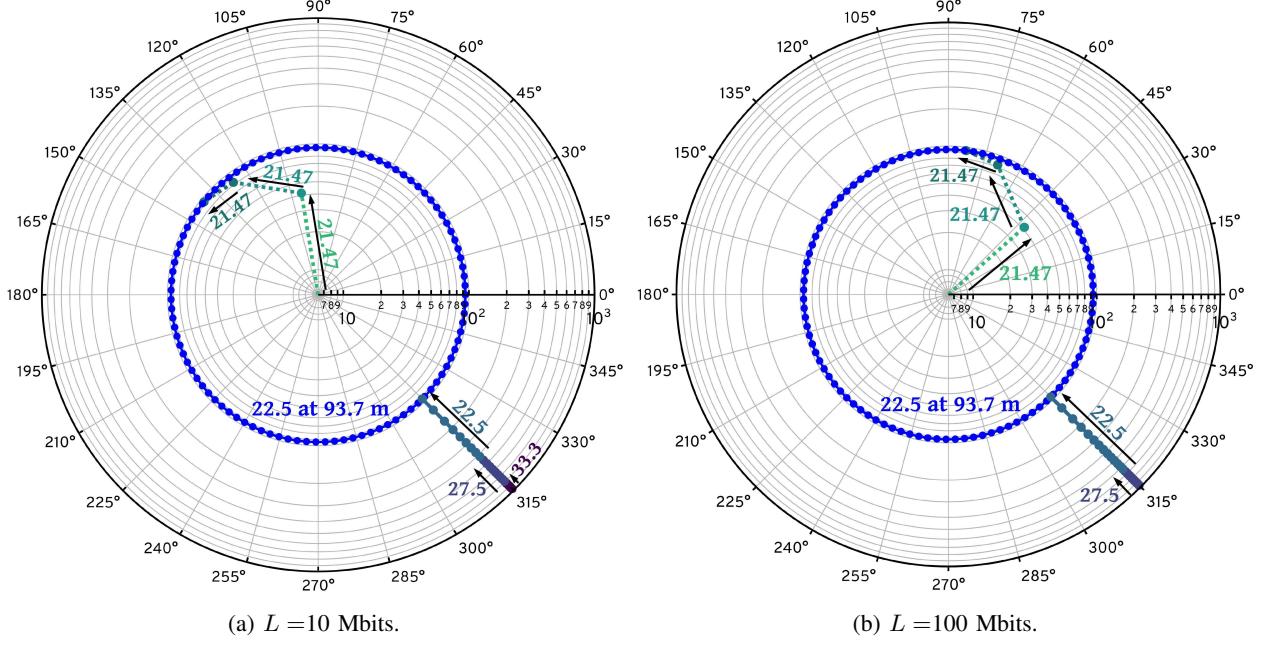


Fig. 10: Optimal Wait Policy for $L = 10$ Mbits ($\Lambda' = 0.2$ req/min/UAV) and $L = 100$ Mbits ($\Lambda' = 0.033$ req/min/UAV): the arrows describe the direction of motion, and the numerical values represent the flying speed in m/s.

the state-of-the-art (SCA [10], CSCA-ADMM [16], DDQN [19], CIRCLE [20]). Correspondingly, charts studying the contributions of communication times and queue wait times are provided in Fig. 11b. We observe that, except for the larger delay incurred to transmit the larger payload, the results are qualitatively similar to Figs. 5a and 5b of the revised manuscript.

Finally, note that the metric optimized by HCSO reflects a trade-off between minimizing the energy expenditure and minimizing the service delay, obtained by changing the value of α . This trade-off is now depicted in Fig. 12 shown in this response.

- 3) **Comment:** *The paper has a lot of redundant equations and well-known trends. The authors had to use the space more efficiently by adding more simulations and verifications.*

R: In our revised manuscript, we have utilized space more efficiently by reorganizing the analytical results into propositions and moving a sketch of their proofs to the Appendix. First, we derive the solution of the optimal throughput-maximizing rate used for rate adaptation in Prop. 1 on page 8, with its proof located in Appendix on page 30. Next, we derive upper and lower bounds to \bar{D}_μ in Prop. 2 on page 12, with its proof located in Appendix on page 30. Finally, derive the steady-state probability π_{comm} that the UAV is in the SMDP state in Prop. 3 on page 15, with its proof located in Appendix on page 31.

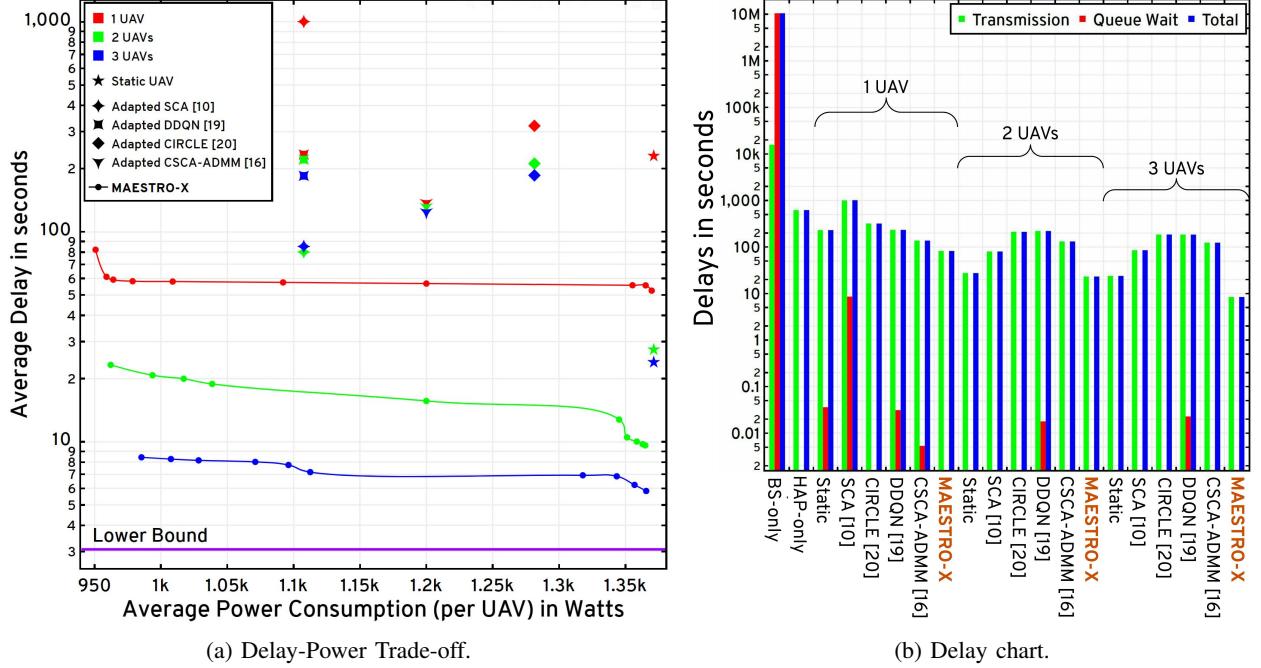


Fig. 11: $L=100$ Mbits, $\Lambda'=0.033$ req/min/UAV: Delay-power trade-off (a) and delay charts (b) for MAESTRO-X, state-of-the-art algorithms, and custom heuristics.

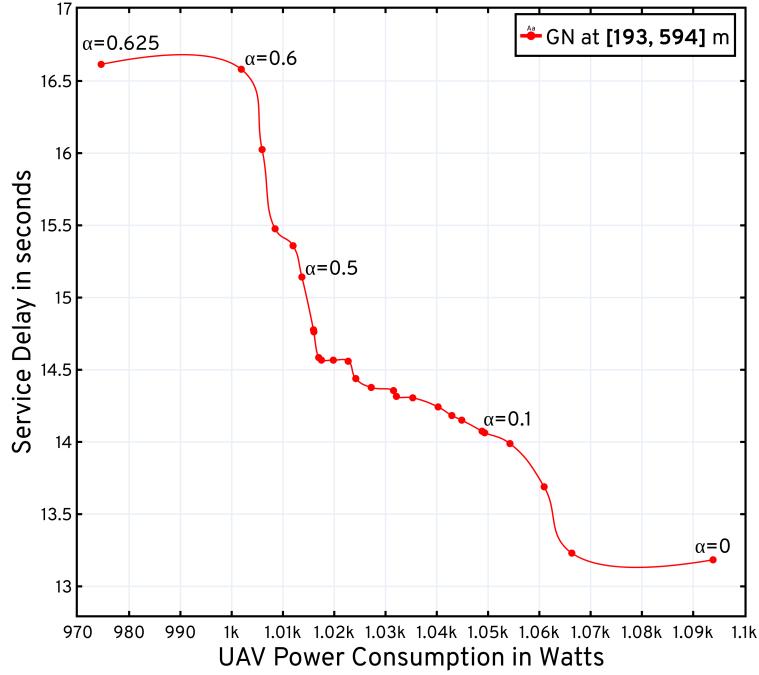


Fig. 12: HCSO Optimal Trajectory Delay-Power Regulation.

Furthermore, we would like to clarify that, although *value iteration* and *competitive swarm optimization* (CSO) are well-known algorithms, used within our proposed MAESTRO-X framework, we do not blindly apply these algorithms to our framework, but specialize them

in a non-trivial way:

- Note that a blind application of value iteration would entail extremely large action and state spaces (trajectories as well as UAV and GN positions), leading to slow convergence times. Instead, we carefully leverage the structure of the problem to optimize the decision-making process. In particular, our careful choice of states and actions leads to the two-scale decomposition of the decision-making process into outer and inner actions. For instance, in communication states, thanks to this decomposition, the outer action optimized via value iteration needs only specify a target end radius during the communication phase (rather than an entire trajectory); given such outer action, the inner action (the actual trajectory of the UAV) can be optimized efficiently (via our newly proposed HCSO) by greedy minimization of a Lagrangian cost metric, capturing the trade-off between delay and energy consumption.
- Note also that the SMDP formulation is highly non-trivial in our setup: it required us to devise an alternative optimization problem (see discussion on page 11, now also validated numerically in Table III), along with a careful definition of states, actions, and when states are sampled in time to obtain the Markovian dynamics of SMDPs (see Sec. III).
- Lastly, note that a direct application of CSO suffers from a complexity-accuracy dilemma: the optimization of high-resolution trajectories outright would be slow to converge, while low-resolution ones would give rise to poor solutions that fail to capture fine-grained variations in the trajectory way-points and velocities that lead to reduced communication delay and power consumption. Instead, to obtain high-resolution trajectories with a small computational footprint, we embed CSO in a hierarchical wrapper: starting from a low-resolution trajectory optimized via CSO, after each CSO iteration, the resulting trajectory is interpolated to form a higher-resolution reference trajectory. The swarm size is reduced, and a new set of particles are generated randomly in the neighborhood of the reference trajectory, in order to preserve the quality of previous lower-resolution trajectories. This process repeats until the desired resolution (number of way-points and speeds) is achieved.

With the space made available in the manuscript as a result of this reorganization, we have added new numerical evaluations: see our response to your Comment 2 for more details on these new simulations.

REVIEWER 3

- 1) **Comment:** *In the Introduction part, the authors compare this work with related works in many aspects. I believe it is better if the authors emphasize the key innovations of this work.*

R: We thank the reviewer for the valuable assessment. In the revised manuscript, we have addressed the lack of clarity in explaining the technical contributions of this work. The primary technical challenge addressed in this work is to create a framework for the *scalable* orchestration of UAV relay swarms. Although state-of-the-art model-free approaches in the research literature, often applying deep reinforcement learning (see [19]–[22]), consider unknown system dynamics in their formulations to solve for the optimal trajectory design and/or scheduling solution, they fail to efficiently capture the problem structure, resulting in slow policy convergence times which limit the scalability of multi-UAV formulations.

Furthermore, we would like to clarify that, although *value iteration* and *competitive swarm optimization* (CSO) are well-known algorithms used within our proposed MAESTRO-X framework, we do not blindly apply these algorithms to our framework, but rather specialize them for single-UAV deployments:

- First, blindly applying value iteration would entail extremely large action (UAV trajectories) and state spaces (UAV and GN positions), leading to slow convergence times. Instead, we carefully leverage the problem structure to optimize the decision-making process. Our careful choice of states and actions leads to the two-scale decomposition of the decision-making process into outer and inner actions. In communication states, thanks to this decomposition, the outer action optimized via value iteration needs only specify a target end radius during the communication phase (rather than an entire trajectory); given such outer action, the inner action (the actual trajectory of the UAV) can be optimized efficiently (via our newly proposed HCSO) by greedy minimization of a Lagrangian cost metric, capturing the trade-off between delay and energy consumption.
- Second, note also that the SMDP formulation is highly non-trivial in our setup: it required us to devise an alternative optimization problem (see discussion on page 11, now also validated numerically in Table III), along with a careful definition of states, actions, and when states are sampled in time to obtain the Markovian dynamics of SMDPs (see Sec. III).
- Lastly, a direct application of CSO suffers from a complexity-accuracy dilemma [26]: the

optimization of high-resolution trajectories outright would be slow to converge, while low-resolution ones would yield poor solutions that fail to capture fine-grained variations in the trajectory way-points and velocities that lead to reduced communication delay and power consumption. Instead, to obtain high-resolution trajectories with a small computational footprint, we embed CSO in a hierarchical wrapper: starting with a low-resolution trajectory optimized via CSO, after each CSO iteration, the resulting trajectory is interpolated forming a higher-resolution reference trajectory. The swarm size is reduced, and a new set of particles are generated randomly in the neighborhood of the reference trajectory, in order to preserve the quality of previous lower-resolution trajectories. This process repeats until the desired resolution (number of way-points and velocities) is achieved.

To extend this single-UAV optimal policy to multi-UAV swarms, we apply policy replication, combined with multi-agent mechanisms enacted over a decentralized command-and-control network (such as spread maximization, consensus-driven conflict resolution, the newly added adaptive frequency reuse and piggybacking mechanisms, described in Sec. V) in order to enable a scalable multi-UAV design.

The revised manuscript now includes a Contributions section in the introduction (see page 5) that explicitly highlights the importance and novelty of these technical contributions. These revisions are quoted below.

"Incorporating waiting state optimization, computationally feasible trajectory design [...] automate the operations of distributed UAV-relay deployments." (Sec. I on page 2)

"We develop a value iteration algorithm [25] exploiting this multiscale structure to optimize outer actions, and a hierarchical variant of Competitive Swarm Optimization (CSO) [26], decoupled from value iteration, to optimize high-resolution trajectories embedding a novel throughput maximizing rate adaptation scheme [...] and piggybacking to enable each UAV to serve multiple users simultaneously." (Sec. I on page 3)

"Contributions: We develop a novel framework for the scalable orchestration of UAV-relay swarms. To the best of our knowledge, no other work simultaneously incorporates the practical features of [...]." (Sec. I on page 5)

Finally, the columns of Table I (Sec. I on page 3) constitute the features necessary for

efficient deployments of UAV-relays in real-world settings to enhance the coverage and service capabilities of a terrestrial BS—namely, dynamic traffic arrivals from randomly located users, simultaneous multi-user service, improving spectrum utilization efficiency, distributed control of UAVs in the swarm, etc. As evident from this table, MAESTRO-X incorporates all these crucial features by solving their corresponding technical challenges, resulting in a scalable, computationally tractable solution for orchestrating a swarm of rotary-wing UAV-relays.

- 2) **Comment:** *This paper assumes that a UAV relay only serves one ground user at a time. It seems that a relay serving multiple users is the more realistic case. Besides, since the number of UAVs is usually much lower than the number of ground users, a major part of users still communicates with the BS directly. This may suggest that the performance gain by utilizing the UAV relays is insufficient.*

R: We thank the reviewer for the valuable assessment, and we agree with the notion that, typically, there are many more GNs than UAVs, hence each UAV should be able to serve multiple requests simultaneously. To address this concern, in the revised manuscript we have added a "piggybacking" mechanism, that allows each UAV to serve multiple users simultaneously (see Sec. V). The piggybacking process is incorporated into the consensus-driven conflict resolution mechanism, whose cost-of-service metrics for choosing to relay a new request through a UAV are detailed in Sec. V. This mechanism allows new requests to be served simultaneously by the UAV, even when the UAV is currently serving requests and following the collation of these previous trajectories (including those of previously piggybacked requests).

We have also realized that, in the original manuscript, the traffic generation rate was too small to justify the use of these newly added mechanisms. Therefore, in the revised manuscript, we have evaluated a new "high-congestion" scenario (see Figs. 6 and 7 and accompanying discussion). In this regime, piggybacking of multiple requests and frequency reuse become necessary to handle the large amount of traffic arriving, hence reduce the queuing delay. In fact, as shown in Fig. 6b for 2 and 3 UAVs, MAESTRO-X performs remarkably well in the high-traffic scenario, with a small queuing delay.

We refer to Sec. V on page 23 for a detailed discussion of the piggybacking mechanism, as well as to Figs. 6 and 7 and accompanying discussion, demonstrating the impact of the

newly added mechanisms.

- 3) **Comment:** *Do UAVs share the same bandwidth channels with the BS? In that case, interference between UAV-user, BS-user and UAV-BS requires careful consideration.*

R: We thank the reviewer for the valuable assessment. In our original manuscript, each UAV and the BS had dedicated channels, and since each Tx-Rx pair was using an orthogonal data channel, there was no need to account for interference. In the revised manuscript, the total application-wide bandwidth (W) is discretized into N_C orthogonal data channels, independent of the number of UAVs and GNs in the deployment. When a new GN request arises in the cell, the server nodes (UAVs and BS) coordinate over the control channels (band-edges of the spectrum) for data channel assignment, cost-of-service computation, and a subsequent consensus on the best server node to handle the new request (Sec. II on page 6). Also, the system model in our original manuscript did not address the need to enhance the spectrum utilization efficiency of the network. In our revised manuscript, to enhance the spectrum utilization efficiency across our network, we have incorporated an adaptive frequency reuse mechanism that allows data channels to be reused by different Tx-Rx pairs, as long as an SINR degradation condition is met (equivalent to a minimum distance condition; Sec V on page 22). Since a data channel can be employed by more than one transmitter in our network, it is important to account for the interference caused by the other transmitters. Along these lines, in Sec. V on page 22, we discuss the setup involved in determining when a data channel can be reused for service by a certain Tx-Rx pair, which involves ensuring that the interference caused to this Rx by the other transmitters is below a preset threshold, and ensuring that the interference caused by this Tx to the other receivers is also below the preset threshold. This interference condition is presented as an SINR degradation constraint in Sec. V, which is further reformulated as a minimum distance condition between Tx-Rx pairs.

Accordingly, we created additional numerical evaluation scenarios to test the effectiveness of the newly added frequency reuse and piggybacking mechanisms:

- We realized that, in our original manuscript, we only simulated $L=1$ Mb data payloads under a low congestion regime ($\Lambda'=1$ req/min), in which spectrum efficiency was not an issue. Enabled by these additional piggybacking and frequency reuse mechanisms, in the revised manuscript we evaluated numerically the performance with a larger data

payload of $L=10$ Mbits under both a low congestion regime (Figs. 5a and 5b in Sec. VI on page 27) as well as a high congestion regime (Figs. 6a and 6b in Sec. VI on page 28). This new regime demonstrates the effectiveness of these mechanisms in handling incoming data traffic and getting it delivered with small queuing delay (see discussion accompanying Figs. 6a and 6b).

- In the revised manuscript, we also evaluate the impact of the number of channels on the average delay performance of 10 UAVs for $L=10$ Mbits under a high congestion regime of $\Lambda=200$ req/min (see Fig. 7 in Sec. VI on page 29 and accompanying discussion). This figure demonstrates that the system achieves good delay performance with only 4 data channels, which are effectively shared among the BS and the 10 UAVs.

In addition to the new numerical results, this comment has been addressed throughout the manuscript as follows.

"Multi-user communication is enabled via OFDMA over a spectrum of bandwidth W , discretized into N_C orthogonal data channels (possibly, obtained by grouping multiple subcarriers together), each with bandwidth $B \triangleq \frac{W}{N_C}$." (Sec II on page 6)

"A fully-connected mesh network overlaying the BS and UAVs enables command-and-control using the band-edges of the allocated spectrum as control channels. Since control packets constitute short frames relative to the large GN-generated data payloads (communicated over data channels), the control operation latencies are neglected. [...] In Sec. V, we also discuss a *frequency reuse* mechanism to improve spectrum utilization efficiency, and a *piggybacking* mechanism allowing the scheduled UAV to serve multiple requests simultaneously." (Sec II on page 6)

Frequency Reuse: To improve the spectrum utilization efficiency, we propose a frequency reuse mechanism, allowing multiple serving nodes (the BS and UAVs) to share the same data channel simultaneously when serving their respective GN requests. [...] In essence, reuse of an occupied data channel is permitted on the condition that the received SNRs of nodes sharing the data channel degrade no more than an acceptable pre-specified threshold permits. Moreover, to make operations of the frequency reuse mechanism more amenable to our problem, which includes UAVs following time-varying trajectories, we equivalently describe this SNR degradation threshold by instead using a minimum distance threshold d_{th} ." (Sec. V on page 22)

- 4) **Comment:** In (12), the authors replace \bar{D}_μ with \bar{W}_μ in the objective function, since both the lower and upper bounds of \bar{D}_μ are increasing functions of \bar{W}_μ . Will such a problem transformation influence the solution to the problem? The authors are advised to verify it through simulations.

R: We have added the suggested numerical evaluation in the revised manuscript. Table III (Sec. VI on page 25) in the manuscript depicts comparisons between \bar{D}_{μ^*} and its lower bound for various data payload sizes L and data traffic arrival rates per UAV Λ' . Recalling that the lower bound $\bar{W}_\mu^{(s)}$ is the objective function replacing \bar{D}_μ in our alternative optimization problem, we observe from Table III that the optimized value $\bar{W}_{\mu^*}^{(s)}$ of this alternative formulation is essentially identical to the expected delay metric \bar{D}_{μ^*} . We have provided the following accompanying discussion in the revised manuscript (see page 24):

"Validation of surrogate optimization problem (9): First, we justify the efficacy of our alternative optimization framework that replaces the original metric \bar{D}_μ with the lower bound $\bar{W}_\mu^{(s)}$. As depicted in Table III, we observe that the optimized value $\bar{W}_{\mu^*}^{(s)}$ of the alternative formulation (9) is practically identical to the expected delay metric \bar{D}_{μ^*} of the original formulation (6), across various data payload sizes (L) and data traffic arrival rates (Λ'). Hence, replacing \bar{D}_μ with its lower bound $\bar{W}_\mu^{(s)}$ as the optimization metric leads to near-optimal solutions. Notably, the surrogate optimization problem (9) is amenable to dynamic programming tools such as value iteration (see Alg. 1) and enables our proposed two-scale policy decomposition that drastically reduces the size of the action space in our SMDP formulation. These tools would not be directly applicable to the original formulation (6) that uses \bar{D}_μ as the optimization objective."

- 5) **Comment:** There exist some clerical errors in the paper. For example, in the equation between (4) and (5), the latter term in the $Q1$ function is mistaken. The paper needs complete proofread.

R: We have addressed and fixed this clerical error in the Marcum Q -function, and carefully proofread the paper. The expression in error was written as

$$h(Z) \triangleq \ln f(Z) + \ln Q_1 \left(\sqrt{2K}, \sqrt{\frac{(K+1)\beta P}{N_0 B \Gamma}} Z \right).$$

We have corrected this expression in Appendix A on page 30 as follows:

$$h(Z) \triangleq \ln f(Z) + \ln Q_1 \left(\sqrt{2K}, \sqrt{\frac{(K+1)N_0B\Gamma}{\beta P}} Z \right).$$

Moreover, we emphasize that this error was strictly typographical. The correct expression was used in our numerical evaluations and has not impacted our simulations.