

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

Bharath Keshavamurthy*, Matthew Bliss[†], and Nicolò Michelusi*

Abstract

In this work, we detail the design of a scalable framework to orchestrate a swarm of rotary-wing Unmanned Aerial Vehicles (UAVs) serving as cellular relays to facilitate beyond line-of-sight connectivity and traffic offloading for ground users, thereby enhancing the coverage and service capabilities of the constituent terrestrial base station. First, restricting our deployment model to a single relay, we develop MAESTRO, a Multiscale Adaptive Energy-conscious Scheduling and TRajjectory Optimization framework, whose goal is to minimize the time-average service delay of ground users, subject to an average UAV power constraint. Formulated as a semi-Markov decision process and equipped with rate adaptation to efficiently exploit air-to-ground channel conditions, we prove that the underlying objective imbibes a multiscale structure to our model and its resultant solution: outer actions involving radial wait velocities and final service positions are established via value iteration to optimize the long-term delay-power trade-off; consequently, inner actions are selected to determine power-minimizing angular wait velocities and service trajectories that minimize an instantaneous delay-power cost metric via hierarchical competitive swarm optimization. Next, this single-agent construction is eXtended to UAV swarms (MAESTRO-X) by replicating the optimal single-relay policy across the swarm, augmented with spread maximization, M/G/x queueing dynamics, and consensus-driven conflict resolution. Numerical evaluations demonstrate that MAESTRO-X significantly improves scalability and performance vis-à-vis average service times and average per-UAV power consumption: simulating a swarm of 3 UAVs, MAESTRO-X delivers the data payload $4\times$ faster than Q-learning, $6\times$ faster than successive convex approximation, and $10\times$ faster than alternating direction method-of-multipliers. Finally, through emulations employing the radio and vehicle control software of the NSF AERPAW platform, we prove the integration and implementation feasibility of our framework.

Index Terms

Unmanned Aerial Vehicles, Markov Decision Processes, Competitive Swarm Optimization

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.

A preliminary version of this manuscript appeared at IEEE ICC 2020 [1].

The source code for this project is available on GitHub [2].

I. INTRODUCTION

With sustained device proliferation, enterprises across sectors have stepped-up their adoption of unmanned aerial vehicles¹ to gather data, survey infrastructure, monitor operations, and automate logistics [3], [4]. Inevitably, this has fostered varied academic research and industrial R&D on drone-augmented beyond line-of-sight connectivity and traffic offloading in cellular networks: the coverage and service capabilities of an extant terrestrial radio access network² are enhanced by the mobility and maneuverability of these autonomous aerial relays [5], [6].

These assets of drone-assisted networks are especially obvious in military environments, public safety scenarios, disaster-relief applications, and mechanized farming. Our prior work on cognitive communications [7] involved performance analyses and mobility studies of an intelligent spectrum sensing and access framework, applied to emulated military scenarios constituting troop deployments accompanied by unmanned aerial relays (e.g., Alleys of Austin scenario evaluations in the DARPA Spectrum Collaboration Challenge [8]). 5G-enabled autonomous drones have facilitated near real-time high-definition video streaming of potential public-safety risks, thereby allowing state and local agencies to effectively monitor and deploy resources to handle these emerging situations with less risk [9]. When the existing wireless infrastructure was rendered inaccessible during the Big Hollow fire in Washington state, Verizon’s fleet of LTE-enabled drones allowed for efficient and expedient deployments of robust air-to-ground links to relay real-time imagery to an operations base several miles away [10]. Recent innovations in sensor manufacturing and virtualization platforms have resulted in their widespread expansion into precision agriculture: unmanned aerial systems serve as additive companions to MooCall livestock monitoring sensors, AgOpenGPS seeding workflows, and VIGOR soil-moisture sensing platforms [11]. Fig. 1 depicts a typical deployment seen in mechanized farms and ranches, where a swarm of UAVs support traffic offloading for cellular devices and extend coverage for precision agricultural sensors.

Unsurprisingly, the pervasive potential of such hybrid wireless networks brings along a plethora of challenges in real-world deployments [6]: specifically, on-board energy constraints of aerial platforms impacting mission times, stringent Quality-of-Service (QoS) requirements for accessible and reliable connectivity, channel stochastics of air-to-ground links in highly-mobile settings,

¹These are also referred to as drones, unmanned aerial systems, and autonomous aerial relays.

²This can either be a single base station or a K-tier heterogeneous network of several base stations.

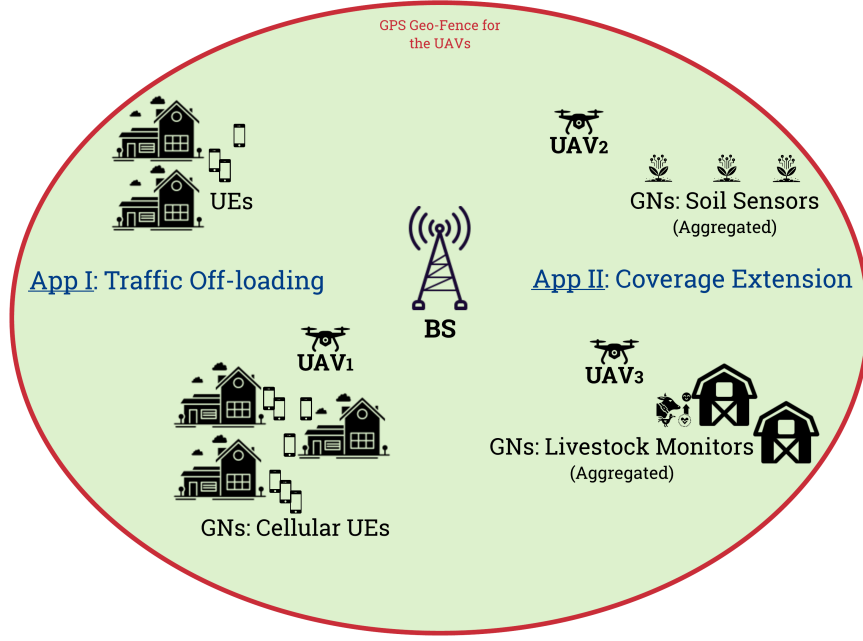


Fig. 1: (*Not to scale*) A terrestrial Base Station (BS) aided by a swarm of 3 Unmanned Aerial Vehicles (UAVs) serving as wireless relays for a diverse set of Ground Nodes (GNs): traffic offloading for cellular User Equipment (UEs) and coverage extensions for livestock monitors and soil sensors. The GPS geo-fence ensures bounded operations of the UAVs.

and computational feasibility challenges in trajectory design brought on by the inherently large state and action spaces. Ergo, several works in the state-of-the-art have tried to tackle these challenges using tools from convex optimization, machine learning, and reinforcement learning—however, various problems remain unsolved and various challenges are left unaddressed.

Recognizing the need to abridge this wide-gap between conceptualization and implementation, we consider a distributed deployment where multiple power-constrained Unmanned Aerial Vehicles (UAVs), equipped with transceiver chains, supplement a terrestrial Base Station (BS) by relaying dynamically-generated data traffic from a random distribution of Ground Nodes (GNs). Coupled with rate adaptation to leverage Air-to-Ground (A2G) propagation conditions, management of queueing dynamics, computing accelerations, and consensus-driven decentralized operations, the proposals detailed in our work constitute a scalable framework to efficiently automate a swarm of UAV relays.

Novelties: Constructing a realistic channel model that accounts for A2G channel characteristics in highly-mobile settings, we propose a computationally efficient method for rate adaptation that optimizes throughput by leveraging large-scale path-loss conditions and small-scale fading statistics. Initially, we constrain our study to a single UAV relay by proposing the Multiscale Adaptive Energy-conscious Scheduling and TRajjectory Optimization (MAESTRO) framework.

In particular, with the underlying objective of minimizing the time-average communication service delay subject to an average UAV mobility power constraint, we formulate the problem as a Semi-Markov Decision Process (SMDP). Considering the UAV's operations in both its waiting and communication states, this formulation and its resultant solution exhibit a multiscale structure, i.e., minimizing the long-term delay-power costs yields outer decisions on radial wait velocities and service positions (via value iteration); next, given these outer decisions, greedily minimizing the instantaneous delay-power costs yields inner actions on angular wait velocities (via exhaustive search) and service trajectories (via hierarchical competitive swarm optimization). In addition to an efficient formulation and smart algorithmic choices, the implementation of MAESTRO is computationally accelerated by capitalizing on multi-threading concurrency and distributed computing capabilities available in software ecosystems.

Post single-relay policy convergence, setting up an eXtension of MAESTRO to UAV swarms (MAESTRO-X), we embed the single-relay optimal policy with suitable heuristics: with each relay disseminating its status and position over a command-and-control network, a spread maximization algorithm is associated with the optimal wait actions to efficiently position and prime the UAVs for potential new service requests; and, maintaining similar exchanges of service times and queue wait delays, a consensus-driven conflict resolution algorithm is associated with the optimal communication service actions, with the goal of choosing the best possible server for the active user request. Finally, by provisioning virtual machines corresponding to the deployment architecture detailed earlier, we demonstrate the API-integration and implementation viability of MAESTRO-X via QGroundControl-based emulations on the NSF AERPAW platform.

A condensed contrast between our approach and those in the current literature on UAV-augmented networks is presented in Table I. Analogous to the columns of Table I, a detailed review follows.

Related Work: First, reviewing single-relay formulations in existing works, we observe non-adaptive schemes [12], [15], [16], [18], [26] designed for applications where the IoT sensors/devices (GNs) possess local storage or aggregation capabilities allowing for deterministic arrivals of data packets. Yet, practical deployments involve dynamically-generated traffic from miscellaneous sets of GNs, each with varying degrees of QoS mandates and technological prowess: for instance, in industrial wireless sensor networks, devices monitoring critical operational parameters generate traffic at random intervals (triggered by safety advisories or emergency system failures) and require highly-reliable near real-time service; and actuators

Paper	Adaptive Control	Channel Model	UAV Trajectory Design: Mobility Velocity	Deployment of UAVs	Multi-UAV Construction	Overall Formulation	Link Layer Model: Schedule Queue
This	Yes	A2G	Dynamic Variable	Distributed	Decoupled	Model-based	Yes Yes
[12]	No	FSPL	Dynamic Variable	Single	-	Model-based	Yes No
[13]	No	A2G	Dynamic Variable	Centralized	Joint	Model-based	Yes No
[14]	No	A2G	Restricted Fixed	Distributed	Joint	Model-free	No No
[15]	No	FSPL	Dynamic Fixed	Single	-	Model-based	Yes No
[16]	Yes	FSPL	Dynamic Variable	Single	-	Model-based	Yes No
[17]	No	FSPL	Restricted Fixed	Distributed	Joint	Model-free	Yes No
[18]	No	A2G	Static -	Single	-	Model-based	No No
[19]	No	A2G	Static -	Distributed	Joint	Model-based	Yes No
[20]	Yes	FSPL	Static -	Distributed	Joint	Model-based	No No
[21]	No	FSPL	Dynamic Fixed	Centralized	Joint	Model-based	Yes No
[22]	No	A2G	Static -	Centralized	Joint	Model-based	No No
[23]	No	A2G	Restricted Fixed	Distributed	Decoupled	Model-free	No No
[24]	Yes	FSPL	Static -	Distributed	Joint	Model-free	No Yes
[25]	No	A2G	Static -	Distributed	Joint	Model-free	No No
[26]	No	A2G	Dynamic Variable	Single	-	Model-based	Yes No

TABLE I: Comparison of our framework with relevant schemes in the literature

handling low-priority tasks generally possess constrained instruction buffers, thus demanding infrequent connectivity to a central processing unit for uninterrupted operations. In contrast to the non-adaptive literature, we consider dynamic traffic generation from a random deployment of sensors and actuators, thereby constructing a control framework that is adaptive to uncertain system dynamics.

Studying state-of-the-art single-relay constructions further, a common approach to solve for optimal service schedules and their associated trajectories has been Successive Convex Approximation (SCA) [12], [15], [16], [26]. Apart from being computationally infeasible to accommodate dynamic data traffic due to their prohibitively large convergence times, SCA approaches rely on first-order Taylor approximations of the objective and the constraints in their optimization problem to enforce convexity, which introduces inaccuracies into the model; additionally, approximations in the data payload delivery constraint preclude the adoption of rate adaptation which allows the transmitters in the network to leverage channel stochastics to maximize their throughputs. Also, these works employ Free Space Path-Loss (FSPL) communication models that fail to account for the A2G channel characteristics inherent in UAV-assisted wireless networks. In our network, in addition to accurately modeling A2G channel characteristics along the UAV-BS and GN-UAV links, and employing rate adaptation at all the transmitters to efficiently exploit said characteristics, there are no such underlying approximations in our formulation.

Pivoting to the trajectory design problem for a single UAV relay, a Competitive Swarm Optimization (CSO) [27] approach is proposed in this paper to bypass the computational infeasibility encountered by SCA-oriented designs. Unlike SCA, which relies on the existence of convex approximations of the objective and the constraints, CSO does not depend on the specific

problem structure to work effectively. Contrary to the limited update scope of Particle Swarm Optimization (PSO) [28], in which particle updates are driven by the individual and the swarm, CSO exhibits superior performance on various large-scale optimization benchmarks [27] since it involves a more efficient update strategy wherein pair-wise competition is invoked between particles, permitting the winners to advance to the next iteration and the loser particles to learn from the winners. Moreover, several works in the literature that employ PSO, either optimize static hovering positions only [29], [30], or enforce impractical path structures (e.g., moving along the circumference of a circle) [31] and velocity restrictions (e.g., constant throughout the mission) [15]. On the other hand, further scaling CSO to higher dimensional trajectory optimization problems by embedding it within a hierarchical wrapper, our solution enables the pragmatic design of optimal series of way-points and velocities of increasingly higher resolution, with no other path structures imposed.

Next, shifting our attention to swarm orchestration frameworks, we find inefficient solutions such as centralized deployments [13], [21], [22] in which an aggregation gateway coordinates the operations of the UAV relays; or either joint multi-relay optimization methods [13], [19], [20] or model-free formulations consisting of combined state and action spaces [14], [17], [24], [25]. Centralized swarm deployments bring in the need for additional capital and operational expenditure; and joint multi-UAV constructions lead to prohibitively large solution spaces resulting in unnecessary overhead in policy convergence times, which when scaled to larger swarms result in intractability. Mindful of such considerations, we present an orchestration framework suitable for distributed deployments of two or more UAVs in a swarm by coupling our single UAV relay policy with spread maximization and consensus-driven link-layer prescient conflict resolution over a common control channel, and replicating it across the swarm thereby establishing multiple decoupled executions of a single optimal policy. This approach eliminates the need for a centralized aggregation center, mitigates the computational overhead and infeasibility encountered by joint multi-relay models, and facilitates a seamless incorporation of the M/G/x Data Link Layer (DLL) queueing dynamics at both the BS and the UAVs into our communication scheduling analyses.

On another note, reviewing these works further, we note the following: the multi-UAV framework described in [13] is non-adaptive and employs SCA—which as described earlier—presents several challenges and involves various impracticalities; [14], [17] impose unreasonable path structures (grid tessellations, directional assumptions) and velocity restrictions (fixed); [24], [25]

only solve for the optimal static positioning of the UAV relays; and [17] studies UAV deployments exclusively for edge computing applications and focuses solely on computational bench-marking while forgoing QoS studies.

Although model-free control schemes in the multi-UAV state-of-the-art (Q-learning, deep Q-networks, or temporal difference learning) [14], [17], [23]–[25] consider unknown system dynamics in their formulations to solve for the scheduling and trajectory optimization problem, they fail to effectively capture the problem structure, typically resulting in slow policy convergence times. In this work, we use a model-based approach, by casting the problem as an SMDP which effectively encapsulates the temporal irregularity associated with the state transitions in UAV-aided hybrid networks: this is evident from the multiscale structure of our optimal policy and its underlying formulation.

Surveying the relevant literature for analyses on queueing dynamics in UAV-augmented wireless networks, we find that the deep-Q network solution proposed in [24] brings in queueing mechanics into its QoS studies. However, unlike our framework in which queueing models are incorporated at the BS and the UAVs, [24] studies queueing at the ground users—which is an impractical assumption for inexpensive IoT sensors/actuators: since the BS and the UAVs are server nodes that are larger and several times more expensive than common GNs, the inclusion of queueing-induced latencies at these server nodes are relatively more important. Furthermore, the model-free approach proposed in [24] suffers from slow policy convergence times.

Finally, we perform emulations of both our standalone and swarm orchestration frameworks on the Experimental Virtual Machines (EVMs) of the NSF AERPAW platform [32] in order to demonstrate the simplicity with which our approach integrates with standard emulation and execution APIs as well as the overall feasibility of our proposal in real-world deployments.

Summarizing our review, we find that—to the best of our knowledge—no other work in the current state-of-the-art develops an orchestration framework for UAV relay swarms while incorporating practical features such as dynamic data traffic from miscellaneous sets of randomly distributed GNs; a scalable and tractable solution to the trajectory design problem, without unrealistic assumptions on UAV mobility; energy-conscious control of the UAV relays during periods of idle operation; efficient exploitation of A2G channel characteristics via rate adaptation, without unreasonable approximations of the objective or the constraints; accelerated policy convergence times via multi-threading concurrency and distributed computing; inclusion of M/G/x queueing dynamics in scheduling decisions; decoupled executions of a single trained policy over

consensus-driven distributed deployments; and finally, pre-deployment emulations on advanced wireless platforms to complete the design path from conceptualization to implementation.

The rest of the paper is organized as follows: Sec. II introduces the system model; Sec. III elucidates the design of MAESTRO, the SMDP formulation for our single-relay construction; Sec. IV describes the algorithms inherent in the policy optimization process of this formulation; Sec. V outlines the development of MAESTRO-X, our extension of the single-relay policy to distributed deployments of two or more UAVs; Sec. VI chronicles our simulation setup and the subsequent numerical evaluations; Sec. VII details NSF AERPAW emulations of MAESTRO-X; and finally, Sec. VIII lists our concluding remarks.

II. THE SYSTEM MODEL

We construct our problem by first motivating the study of conventional radio ecosystems aided in their coverage and service capabilities by unmanned aerial relays. From a coverage extension perspective, it is evident that due to poor path-loss conditions experienced by ground users farther away from the serving base station (especially, inexpensive and low-power IoT devices) and signal attenuations due to non-line-of-sight propagation (due to beyond visual line-of-sight links or blockages), autonomous aerial platforms serving as cellular relays augment the coverage capabilities of the serving base station³. From a traffic offloading perspective, these aerial relays assist in handling requests that arise when service through the base station is backlogged.

Deployment Model: Under this context, consider a generalized scenario in which a swarm of N_U rotary-wing Unmanned Aerial Vehicles (UAVs)⁴—each equipped with an on-board transceiver chain—operate as cellular relays to supplement the coverage and service capabilities of a terrestrial Base Station (BS) by relaying data traffic dynamically-generated by users at ground level, i.e., Ground Nodes (GNs). An example of one such scenario is depicted in Fig. 1. The BS is located at the center of the circular cell (of radius a), at height H_B , while the UAVs operate at a fixed height of H_U . The GNs are distributed uniformly at random throughout the cell, with a density of λ_G [GNs per unit area]. The BS utilizes k orthogonal channels to serve the GNs simultaneously via an Orthogonal Frequency Division Multiple Access (OFDMA) strategy; on the other hand, the UAV relays are restricted to serve one GN at a time through a decode-and-

³This benefit of aerial relays is particularly obvious in millimeter-wave communications [33].

⁴The proposed solutions in this paper can be applied to fixed-wing UAVs [34] by employing appropriate mobility power models and suitable flying velocity constraints in the underlying formulation.

forward scheme⁵. All channels are assumed to have a bandwidth of B . Without any loss of generality, studying uplink transmissions only⁶, these GNs generate random data traffic that is to be transmitted to the BS, either directly or by using one of the UAVs as a relay.

Communication Model: Each GN generates uplink transmission requests of L bits, according to a Poisson process with rate $\lambda_{R|G}$ [requests per GN per unit time]. Coupled with the random deployment of GNs, uplink transmission requests arrive in time according to a Poisson process with rate $\lambda_R = \lambda_G \cdot \lambda_{R|G}$ [requests per unit time per unit area]. Thus, $\Lambda \triangleq \lambda_R \pi a^2$ [requests per unit time] is the overall request arrival rate over the circular cell. Since a new request is uniformly distributed in the cell area, its angular coordinate θ is uniform in $[0, 2\pi)$ and the probability density function of its radial coordinate is $f_R(r) = \frac{2r}{a^2} \mathbb{I}(r \leq a)$, where $\mathbb{I}(\cdot)$ is the indicator function. A fully-connected mesh network is overlaid on the BS and the UAVs in the cell in order to establish a command-and-control network, allocating the band-edges of the spectrum under use as control channels. Since the packets exchanged among the mesh nodes over the control channel constitute short frames relative to the large data payloads generated by the GNs (and communicated over orthogonal data channels), it is reasonable to neglect the latencies involved in these control operations. When a GN decides to upload its data, it informs the BS—over the control channel—about the need for an uplink transmission of L bits, and includes its physical location in this preliminary request for service. Considering potential delay-power costs for this request, the BS and the UAVs coordinate over the control network to arrive at a consensus on the best scheduling decision: if direct transmission is chosen, the BS assigns a data channel $k \in \{1, 2, \dots, N_B\}$ to the GN and instructs it to begin transmission; else, if relaying the data payload through UAV i is determined to be the most efficient choice, the UAV instructs the GN to begin transmission over its designated pre-determined data channel⁷ $k_{U_i} \in \{1, 2, \dots, N_U\}$. A *Decode-and-Forward*⁸ (D&F) strategy underlies the communication process encountered in the latter case: while moving along a designed trajectory (a sequence of way-points and velocities), UAV i first receives the entire

⁵Along with supplementary DLL heuristics in Sec. V, our analyses on the impact of queueing dynamics at both the BS as well as the UAVs are presented in Sec. VI.

⁶This model can be extended to both uplink and downlink transmissions, by creating an additional state differentiating the two types of data traffic: when a downlink request is generated, the BS may choose to either transmit the downlink data payload directly to the destination GN or relay it through one of the UAVs in the swarm.

⁷Although our communication model assumes pre-assigned channels with fixed bandwidths, our previous work on intelligent channel and bandwidth allocations [7] can be adopted here with minimal changes to the subsequent constructions.

⁸An *Amplify-and-Forward* strategy [35] can also be employed in our construction by allowing the UAVs to possess two on-board transceiver chains—capable of simultaneous operations—with each chain assigned a pre-determined data channel.

data payload from the GN over channel k_{U_i} (*decode*) and subsequently transmits it to the BS over the same channel (*forward*). As a result of these different scheduling decisions, the GN→BS, GN→UAV, and the UAV→BS links must be studied. We discuss them 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 $\mathbb{E}[|g|^2]=1$ is the small-scale fading component. We model the large-scale component as $\beta = \beta_{\text{LoS}}(d) \triangleq \beta_0 d^{-\alpha}$ for line-of-sight (LoS) and $\beta = \beta_{\text{NLoS}}(d) \triangleq \kappa \beta_0 d^{-\tilde{\alpha}}$ for non-LoS (NLoS) links, where β_0 is the pathloss referenced at a distance of 1 meter, $2 \leq \alpha \leq \tilde{\alpha}$ are the LoS and NLoS path-loss exponents, $\kappa \in (0, 1]$ captures the additional NLoS attenuation, and d is the Euclidean distance between the transmitter (Tx) and the receiver (Rx) [12]. We model the LoS and NLoS probability as a function of the elevation angle $\varphi \in (0, 90^\circ]$ as [36]

$$P_{\text{LoS}}(\varphi) = \frac{1}{1 + z_1 \exp(-z_2 [\varphi - z_1])}, \quad P_{\text{NLoS}}(\varphi) = 1 - P_{\text{LoS}}(\varphi), \quad (1)$$

where z_1 and z_2 are parameters specific to the propagation environment, e.g., rural, suburban, urban, etc. The distribution of the small-scale fading component g also depends on the LoS or NLoS link state [37]. Specifically, for a LoS link, as in [26], we model g as Rician fading with a φ -dependent K -factor, i.e., $K(\varphi) = k_1 \exp(k_2 \varphi)$, where coefficients k_1 and k_2 are determined by the propagation environment [26]. For a NLoS link, we model g as Rayleigh fading [37] (Rician with $K=0$). Given h , the link capacity is $C(h) = B \cdot \log_2 \left(1 + \frac{|h|^2 P}{N_0 B \Gamma} \right)$, where P is the transmission power, N_0 is the noise power spectral density at the receiver, B is the channel bandwidth, and Γ is the Signal-to-Noise Ratio (SNR) gap between practical modulation-and-coding schemes and theoretical Gaussian signaling [26]. We assume that other sources of signal degradation, such as the Doppler effect, are well-compensated at the receiver, using techniques developed in [38]. Since the large-scale components typically vary slowly relative to the rate of acquisition 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 process⁹, which enables rate-control at the transmitter; on the other hand, small-scale fading conditions vary at a much faster timescale, hence cannot be tracked at the transmitter, which may result in outages when the selected rate exceeds the channel capacity $C(h)$. Thus, given (β, K) and a transmission rate of

⁹Note that β and K vary due to temporal variations in the LoS or NLoS propagation conditions, and in the Tx-Rx distance d and elevation angle φ .

Υ [bits per second], we define the outage probability $P_{\text{out}}(\Upsilon, \beta, K) \triangleq \mathbb{P}(C(\sqrt{\beta}g) < \Upsilon | \beta, K) = \mathbb{P}(|g|^2 < u(\Upsilon, \beta))$, where $u(\Upsilon, \beta) \triangleq N_0 B \Gamma (2^{\Upsilon/B} - 1) / (\beta P)$. 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

$$P_{\text{out}}(\Upsilon, \beta, K) = 1 - Q_1\left(\sqrt{2K}, \sqrt{2(K+1)u(\Upsilon, \beta)}\right), \quad (2)$$

where $Q_1(\cdot, \cdot)$ is the standard Marcum Q -function [26]. Note that when $K=0$ (Rayleigh fading NLoS link), the function specializes to $Q_1\left(0, \sqrt{2u(\Upsilon, \beta)}\right) = \exp(-u(\Upsilon, \beta))$. We assume that the small-scale fading is averaged out in the temporal and spatial dimensions, yielding the expected throughput as

$$R(\Upsilon, \beta, K) = \Upsilon \cdot (1 - P_{\text{out}}(\Upsilon, \beta, K)) = \Upsilon \cdot Q_1\left(\sqrt{2K}, \sqrt{2(K+1)u(\Upsilon, \beta)}\right). \quad (3)$$

In our model, we permit rate adaptation at the transmitter based on the large-scale parameters (β, K) , coordinated through the control channel via CSI feedback. In particular, the transmission rate Υ is chosen to maximize the expected throughput given (β, K) , yielding

$$\Upsilon^*(\beta, K) \triangleq \arg \max_{\Upsilon \geq 0} R(\Upsilon, \beta, K).$$

To solve this problem, let $Z \triangleq \sqrt{\frac{2\beta P}{N_0 B \Gamma} u(\Upsilon, \beta)}$, so that $\Upsilon = B \log_2(1 + \frac{1}{2}Z^2) \triangleq f(Z)$. It follows that $\Upsilon^*(\beta, K) = f(Z^*(\beta, K))$, where $Z^*(\beta, K) \triangleq \arg \min_{Z \geq 0} g(Z)$, where $g(Z) \triangleq -\ln f(Z) - \ln Q_1\left(\sqrt{2K}, \sqrt{\frac{(K+1)\beta P}{N_0 B \Gamma}} Z\right)$. Since the function $g(Z)$ is convex, $Z^*(\beta, K)$ can be found efficiently using a bisection method. Upon determining the optimal transmission rate $\Upsilon^*(\beta, K)$, we define the optimized throughput, as a function of the large-scale conditions, as $R^*(\beta, K) \triangleq R(\Upsilon^*(\beta, K), \beta, K)$. Assuming that the LoS and NLoS conditions are averaged out in the temporal and spatial dimensions, we compute the average link throughput coupled with rate adaptation as

$$\bar{R}(d, \varphi) \triangleq P_{\text{LoS}}(\varphi) \cdot R^*(\beta_{\text{LoS}}(d), K(\varphi)) + P_{\text{NLoS}}(\varphi) \cdot R^*(\beta_{\text{NLoS}}(d), 0), \quad (4)$$

which is then specialized to the three distinct communication links by expressing the transmission powers, the environment-specific parameters z_1, z_2, k_1 , and k_2 , the large-scale parameters (β, K) , and the LoS or NLoS probabilities (1) based on the spatial configuration, i.e., Tx-Rx distance and elevation angle. Specifically, for the GN \rightarrow 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 $\varphi = \sin^{-1}(H_B/d)$ in (4). Similarly, for the GN \rightarrow 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 $\varphi = \sin^{-1}(H_U/d)$ in (4). Finally, for the UAV \rightarrow 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 $\varphi = \sin^{-1}((H_U - H_B)/d)$ in (4).

UAV Mobility Power Model: Let $P_{U_i}(t) = P_{U_i, \text{comm}}(t) + P_{\text{mob}}(v_{U_i}(t))$ be the instantaneous power consumption of UAV i comprising the total communication power $P_{U_i, \text{comm}}(t)$ and the forward flight mobility power $P_{\text{mob}}(v_{U_i}(t))$, a generally non-convex function of the UAV horizontal flying velocity $v_{U_i}(t)$, the vehicle drag coefficient, and the total vehicle in-flight weight [39]. Since the communication power (≈ 5 W) is usually dwarfed by the mobility power (≈ 1000 W), we approximate $P_{U_i}(t) \approx P_{\text{mob}}(v_{U_i}(t))$ as [12]

$$P_{\text{mob}}(V) = P_1 \left(1 + \frac{3V^2}{U_{\text{tip}}^2} \right) + P_2 \left(\sqrt{1 + \frac{V^4}{4v_0^4}} - \frac{V^2}{2v_0^2} \right)^{1/2} + P_3 V^3, \quad (5)$$

where $v_{U_i}(t) = V$, $P_j, j \in \{1, 2, 3\}$ are scaling constants, U_{tip} is the rotor blade tip velocity, and v_0 is the mean rotor induced velocity while hovering (see Sec. VI for the simulation values of these constants). This model is validated by real UAV deployments [39] and a power consumption versus velocity curve found in [12]. Therein, hovering requires 1370 W, whereas flying at the power-minimizing velocity of 22 m s^{-1} only consumes 940 W. This suggests that, as we will show, the mobility of the UAVs can be exploited to reduce their power consumption, while simultaneously ensuring that the QoS mandates for the GNs in the cell are successfully met.

Along these lines, the objective of the MAESTRO-X framework proposed in this paper is to solve for an energy-conscious adaptive service scheduling and trajectory optimization scheme wherein a distributed swarm of UAV relays are employed to minimize the time-average communication service delay encountered by GNs in the cell, randomly generating uplink transmission requests, subject to an average per-UAV mobility power constraint. Note that the delay to serve a request from a GN in position (r, θ) by direct transmission to the BS is $\frac{L}{R_{GB}(r)}$. Then, if all requests are served directly by the BS, the expected delay is obtained by computing the expectation with respect to the radial coordinate, i.e., $\bar{D}_{BS} \triangleq \int_0^a \frac{L}{R_{GB}(r)} f_R(r) dr$. Thus, mindful of the poor path-loss conditions experienced by GNs farther away from the BS and signal attenuations due to NLoS propagations, a quantitative view of the objective would involve orchestrating this autonomous aerial relay swarm to improve the service latencies encountered

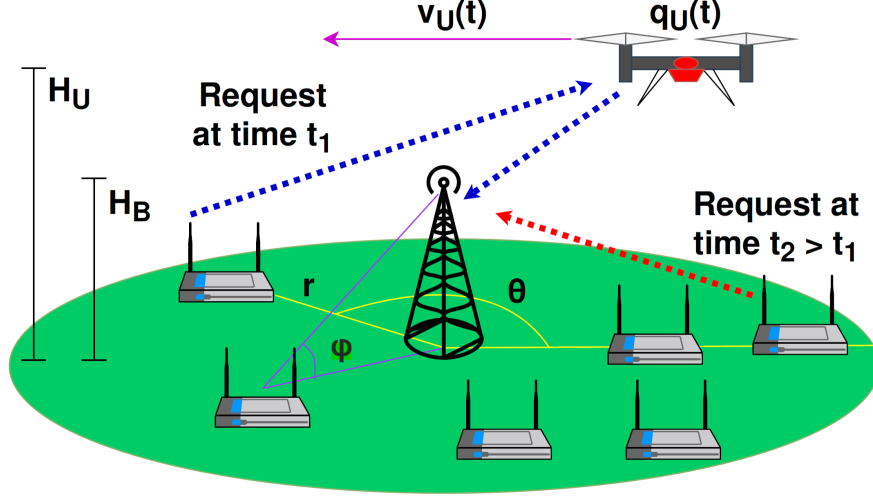


Fig. 2: A specialization of the generic deployment model discussed in Sec. II: a BS augmented by a single UAV serving as a wireless relay for a random distribution of GNs that are dynamically generating uplink transmission requests according to a Poisson process with rate $\frac{\Lambda}{N_U}$ [requests per unit time].

by the GNs in the cell. To this end, in the next section, we formulate a framework specialized to single-UAV deployments.

III. MAESTRO: A SEMI-MARKOV DECISION PROCESS FORMULATION

In this section, studying the setup illustrated in Fig. 2, we specialize the deployment model described in Sec. II to the case of a single UAV, and present MAESTRO, a Multiscale Adaptive Energy-conscious Scheduling and TRajjectory Optimization framework to minimize the time-average service delay subject to an average UAV mobility power constraint via a Semi-Markov Decision Process (SMDP) formulation. Since we will later extend this model to a swarm of N_U UAVs, each scheduled to serve a portion of the traffic, the traffic rate experienced by a single UAV is $\frac{\Lambda}{N_U}$ [requests per unit time], which is assumed in this section in place of the overall rate Λ . We let $\mathbf{q}_U(t) = (r_U(t), \theta_U(t))$ be the polar coordinate of the UAV at time t , projected on 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 (cell center).

Analyzing this model, we note that the operations of the UAV relay can be split into the following phases. In the *waiting* phase, no GN requests are being served by the UAV, which thus moves according to a waiting policy as described next, until a new request is received. When a new GN request is received, say from position (r, θ) , the system transitions to the *request scheduling* phase, where the system decides if the GN should transmit its data payload directly to the BS, or relay it through the UAV. If direct transmission is selected, 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 GN relays its data payload through the UAV using the D&F protocol described in further detail in this section; upon the completion of this relay service, the system re-enters the *waiting* phase. Here, we simplify our construction by neglecting the waiting times encountered by requests in the BS and UAV queues—specifically, in the *request scheduling* phase, the decision-making scheme takes into account only the communication service times in order to optimally choose between direct-BS and relayed service actions. However, in Sec. V, we supplement our optimal single-relay decision making scheme with studies on M/G/x queueing dynamics (at both the BS and the UAVs) to incorporate queue wait latencies into this scheduling decision. As the BS can accommodate simultaneous transmissions (see Sec. II), new requests received during an *UAV relay* phase (i.e., when the UAV is occupied serving another request) are directly served by the BS.

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; let $\Delta_u^{(w)}$ be the time to wait for a new request, and $\Delta_u^{(s)}$ be the time to serve it, either through the BS (denoted with the scheduling decision $\xi_u=0$) or through the UAV ($\xi_u=1$). Then, the u th decision interval duration is $\Delta_u = \Delta_u^{(w)} + \xi_u \Delta_u^{(s)}$, since the UAV enters the *waiting phase* immediately if a direct transmission to the BS is scheduled.

We now formulate the average communication delay and UAV mobility power. 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 additional requests are served directly by the BS, let $\Delta_{u,i}^{(bs)}$, $i = \{1, \dots, N_u\}$ be their communication delays. Let E_u be the UAV mobility energy expended during the u th decision interval. Let M_t be the total number of decision intervals completed up to time t . Then, we define the expected average communication delay per GN request under a given policy μ that defines the request scheduling, communication strategy, and UAV trajectory (formally defined later)¹⁰, as the total communication delay accrued until time t , over the total number of requests served until time t ; similarly, we define the expected average UAV power as the total energy

¹⁰In the following, expectations $\mathbb{E}_\mu[\cdot]$ will be implicitly assumed to be conditioned on an arbitrary initial position $\mathbf{q}_U(0)$ of the UAV at time $t=0$.

consumption, over the total duration of the M_t decision intervals up to time t ; mathematically,

$$\bar{D}_\mu \triangleq \lim_{t \rightarrow \infty} \mathbb{E}_\mu \left[\frac{\frac{1}{M_t} \sum_{u=1}^{M_t} (\Delta_u^{(s)} + \xi_u \sum_{i=1}^{N_u} \Delta_{u,i}^{(bs)})}{\frac{1}{M_t} \sum_{u=1}^{M_t} (1 + \xi_u N_u)} \right], \quad \bar{P}_\mu \triangleq \lim_{t \rightarrow \infty} \mathbb{E}_\mu \left[\frac{\frac{1}{M_t} \sum_{u=1}^{M_t} E_u}{\frac{1}{M_t} \sum_{u=1}^{M_t} \Delta_u} \right]. \quad (6)$$

Note that \bar{D}_μ in (6) 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. The goal is to solve¹¹

$$\bar{D}_\mu^* = \min_{\mu} \bar{D}_\mu \text{ s.t. } \bar{P}_\mu \leq P_{\text{avg}}, \text{ with optimal policy } \mu^*. \quad (7)$$

To simplify these expressions, 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 to denote the long-term average cost C_u per decision interval, and let

$$\bar{E}_\mu \triangleq \bar{\mathbb{E}}_\mu[E_u], \quad \bar{T}_\mu \triangleq \bar{\mathbb{E}}_\mu[\Delta_u], \quad \bar{N}_\mu \triangleq \bar{\mathbb{E}}_\mu[1 + \xi_u N_u], \quad \bar{W}_\mu^{(s)} \triangleq \bar{\mathbb{E}}_\mu[\Delta_u^{(s)}], \quad \bar{W}_\mu^{(bs)} \triangleq \bar{\mathbb{E}}_\mu \left[\xi_u \sum_{i=1}^{N_u} \Delta_{u,i}^{(bs)} \right] \quad (8)$$

be the expected average UAV energy expenditure (\bar{E}_μ), duration (\bar{T}_μ), number of requests served (\bar{N}_μ), delay of requests for which a scheduling decision is made ($\bar{W}_\mu^{(s)}$), and total delays of requests served directly by the BS during the UAV relay phase ($\bar{W}_\mu^{(bs)}$), per decision interval, under policy μ . We can then write $\bar{P}_\mu = \frac{\bar{E}_\mu}{\bar{T}_\mu}$ and $\bar{D}_\mu = \frac{\bar{W}_\mu^{(s)} + \bar{W}_\mu^{(bs)}}{\bar{N}_\mu}$ (Little's Law [40]), yielding the equivalent optimization problem

$$\bar{D}_\mu^* = \min_{\mu} \frac{\bar{W}_\mu^{(s)} + \bar{W}_\mu^{(bs)}}{\bar{N}_\mu} \text{ s.t. } \bar{E}_\mu - P_{\text{avg}} \bar{T}_\mu \leq 0. \quad (9)$$

Under the optimal policy, $\bar{D}_\mu^* \leq \bar{D}_{BS}$ (in fact, a feasible policy is obtained by serving all requests directly by the BS while the UAV flies at the power minimizing velocity). This suggests that optimization of the UAV trajectory and scheduling decisions is key to improving the delay performance to serve requests. Note the inherent complexity that is required to solve the optimization problem (9): as the policy varies, the delay metric changes both the numerator and denominator terms of the objective function, which cannot be solved with standard dynamic programming algorithms. To address this challenge, we now propose an alternative optimization criterion based on upper- and lower-bounds to \bar{D}_μ (Lemma 1, whose proof is provided in Appendix A).

¹¹Without loss of generality, we assume $P_{\text{avg}} \geq \min_{V \geq 0} P_{\text{mob}}(V) \triangleq P_{\text{min}}$, since any $P_{\text{avg}} < P_{\text{min}}$ is not feasible.

Lemma 1. Under any policy μ such that $\bar{W}_\mu^{(s)} \leq \bar{D}_{BS}$ (including the optimal μ^*), we have that

$$\bar{W}_\mu^{(s)} \leq \bar{D}_\mu \leq \bar{W}_\mu^{(s)} \frac{1 + \Lambda \bar{D}_{BS}}{1 + \Lambda \bar{W}_\mu^{(s)}} \leq \bar{D}_{BS}. \quad (10)$$

Remarkably, the lower and upper bounds of \bar{D}_μ are both increasing functions of $\bar{W}_\mu^{(s)}$; hence, we will use $\bar{W}_\mu^{(s)}$ as a surrogate to minimize the expected delay \bar{D}_μ . This observation motivates us to look at the alternative optimization problem

$$\min_{\mu} \bar{W}_\mu^{(s)} \text{ s.t. } \bar{E}_\mu - P_{\text{avg}} \bar{T}_\mu \leq 0, \quad (11)$$

which will be the focus of the following analysis. To solve it, we use the Lagrangian formulation

$$g(\nu) = \min_{\mu} (\bar{W}_\mu^{(s)} + \nu(\bar{E}_\mu - P_{\text{avg}} \bar{T}_\mu)) = \min_{\mu} \bar{\mathbb{E}}_\mu [\Delta_u^{(s)} + \nu(E_u - P_{\text{avg}} \Delta_u)], \quad (12)$$

where ν is the dual variable, optimized by solving $\max_{\nu \geq 0} g(\nu)$. We now demonstrate that for a given $\nu \geq 0$, (12) can be cast as a Semi-Markov decision process (SMDP) and solved with dynamic programming tools. We first characterize the states, actions, and policy of this SMDP.

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

Waiting states' actions, transitions, and metrics: If the UAV is in the *waiting* state $\mathbf{s}_n = \mathbf{q}_U \in \mathcal{S}_{\text{wait}}$ at time t , i.e., it is in position $\mathbf{q}_U(t) = \mathbf{q}_U = (r_U, \theta_U)$ with no active requests, then the actions available are to move the UAV with radial (referred to the outward direction) and angular (referred counter-clockwise) velocity components (v_r, θ_c) , over an arbitrarily small duration $\Delta_0 \ll 1/\Lambda$. Under a maximum velocity constraint V_{max} , the action space is then

$$\mathcal{A}_{\text{wait}}(r_U) \triangleq \left\{ (v_r, \theta_c) \in \mathbb{R}^2 \mid \sqrt{v_r^2 + r_U^2 \cdot \theta_c^2} \leq V_{\text{max}} \right\}, \quad (13)$$

where $v_U = \sqrt{v_r^2 + r_U^2 \theta_c^2}$ is the velocity expressed with respect to 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 visit is $T(\mathbf{s}; \mathbf{a})=\Delta_0$, during which the UAV uses an amount of energy $E(\mathbf{s}; \mathbf{a})=\Delta_0 P_{\text{mob}}(v_U)$ 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) , so that the new state is a communication state. Thus, the transition probability from the waiting state $\mathbf{s}_n=(\mathbf{q}_U, \mathbf{w}_0)$ under action $\mathbf{a}_n=(v_r, \theta_c) \in \mathcal{A}_{\text{wait}}(r_U)$ is

$$\mathbb{P}(\mathbf{s}_{n+1} = \mathbf{q}_U + \mathbf{a}_n\Delta_0 | \mathbf{s}_n, \mathbf{a}_n) = e^{-\Lambda\Delta_0}, \quad (14)$$

$$\mathbb{P}(\mathbf{s}_{n+1} = (\mathbf{q}_U + \mathbf{a}_n\Delta_0, \mathbf{q}'_G) \text{ with } \mathbf{q}'_G \in \mathcal{F} | \mathbf{s}_n, \mathbf{a}_n) = \frac{A(\mathcal{F})}{\pi a^2} \cdot (1 - e^{-\Lambda\Delta_0}), \quad \forall \mathcal{F} \subseteq \mathcal{Q}_{\text{GN}}, \quad (15)$$

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 $\xi=0$, the GN transmits directly to the BS, and the next state is sampled immediately after the decision, so that it is the waiting state $\mathbf{s}_{n+1}=\mathbf{q}_U \in \mathcal{S}_{\text{wait}}$ with probability 1. In this case, the cost metrics under action $\mathbf{a}_n=\mathbf{a}=(0, [\])$ are computed as $\Delta(\mathbf{s}_n; \mathbf{a})=\frac{L}{\bar{R}_{GB}(r)}$, $E(\mathbf{s}_n; \mathbf{a})=0$, $T(\mathbf{s}_n; \mathbf{a})=0$, since direct transmissions occur at throughput $\bar{R}_{GB}(r)$ and the system moves immediately to the waiting state $\mathbf{q}_U \in \mathcal{S}_{\text{wait}}$ resulting in the action duration and energy expenditure being 0.

On the other hand, if $\xi=1$, the UAV uses the D&F protocol described next, while following a trajectory starting from its current position \mathbf{q}_U and ending in position \mathbf{q}'_U . We denote this action as $\mathbf{a}_n=\tilde{\mathbf{a}}=(1, \mathbf{q}_U \rightarrow \mathbf{q}'_U)$. In the first phase (of duration t_p) of the D&F protocol, the GN transmits its payload to the UAV; in the second phase (of duration $\Delta-t_p$), the UAV relays the data payload to the BS. Assuming a *move-and-transmit* implementation [12], the trajectory $(\mathbf{q}_U \rightarrow \mathbf{q}'_U)$ and the durations $(t_p$ and $\Delta-t_p)$ must satisfy

$$\int_0^{t_p} \bar{R}_{GU}(r_{GU}(t+\eta))d\eta \geq L, \quad \int_{t_p}^{\Delta} \bar{R}_{UB}(r_{UB}(t+\eta))d\eta \geq L, \quad (16)$$

i.e., the entire payload of L bits is first transmitted to the UAV with rate $\bar{R}_{GU}(r_{GU}(t+\eta))$, where $r_{GU}(t+\eta)$ is the GN-UAV distance (projected onto the x - y plane) at time $t+\eta$; then, the UAV transmits the payload to the BS with rate $\bar{R}_{UB}(r_{UB}(t+\eta))$, where $r_{UB}(t+\eta)$ is the

radial position of the UAV at time $t+\eta$, so that the total communication delay is Δ . In this case, the cost metrics under action $\tilde{\mathbf{a}}$ are $\Delta(\mathbf{s}_n; \tilde{\mathbf{a}})=\Delta$, $E(\mathbf{s}_n; \tilde{\mathbf{a}})=\int_0^\Delta P_{\text{mob}}(v_U(t+\eta))d\eta$, $T(\mathbf{s}_n; \tilde{\mathbf{a}})=\Delta$. Upon completing the D&F protocol, the UAV enters the *waiting* phase again, so that $\mathbf{s}_{n+1}=\mathbf{q}'_U$ becomes the new SMDP state, sampled at time $t+\Delta$. Let $\mathcal{Q}_{\mathbf{q}_G}(\mathbf{q}_U \rightarrow \mathbf{q}'_U)$ be the set of feasible UAV trajectories starting in \mathbf{q}_U , terminating in \mathbf{q}'_U , to serve a GN located at \mathbf{q}_G via the D&F protocol, i.e.,

$$\mathcal{Q}_{\mathbf{q}_G}(\mathbf{q}_U \rightarrow \mathbf{q}'_U) = \left\{ \mathbf{p}_U : [0, \Delta] \mapsto \mathbb{R}_+ \times [0, 2\pi) \text{ s.t.} \right. \quad (17)$$

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

$$v_U(\eta) \leq V_{\max}, \quad \forall \eta \in [0, \Delta], \quad (C.2)$$

$$\mathbf{p}_U(0) = \mathbf{q}_U, \mathbf{p}_U(\Delta) = \mathbf{q}'_U, \quad \exists \Delta \geq 0, \exists 0 \leq t_p \leq \Delta \Big\}, \quad (C.3)$$

where C.1 reflects the data payload constraints, C.2 the maximum velocity constraint, and C.3 the 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 the *communication* states $(\mathbf{q}_U, \mathbf{q}_G) \in \mathcal{S}_{\text{comm}}$ is then $\mathcal{A}_{\text{comm}}(\mathbf{q}_U, \mathbf{q}_G) \triangleq \{0, [\]\} \cup \{1\} \times \mathcal{Q}_{r,\theta}(r_U, \theta_U)$. Here the set $\{0, [\]\}$ is associated with the decision $\xi=0$ (direct-BS) and hence, there is no trajectory design space corresponding to it; while the set $\{1\} \times \mathcal{Q}_{r,\theta}(r_U, \theta_U)$ is associated with the decision $\xi=1$ (relayed service) whose trajectory design space is $\mathcal{Q}_{r,\theta}(r_U, \theta_U)$.

Policy μ : For *waiting* states $\mathbf{q}_U \in \mathcal{S}_{\text{wait}}$, the policy selects a velocity (v_r, θ_c) from the action space defined in (13), i.e., $\mu(\mathbf{q}_U) \in \mathcal{A}_{\text{wait}}(r_U)$. 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., $\mu(\mathbf{q}_U, \mathbf{q}_G) \in \mathcal{A}_{\text{comm}}(\mathbf{q}_U, \mathbf{q}_G)$.

With a stationary policy μ defined, the Lagrangian metric $L_\mu^{(\nu)} \triangleq \bar{W}_\mu^{(s)} + \nu(\bar{E}_\mu - P_{\text{avg}}\bar{T}_\mu)$ in (12) is reformulated using Little's Law [40] in terms of the SMDP steady-state probabilities as

$$L_\mu^{(\nu)} = \lim_{N \rightarrow \infty} \mathbb{E}_\mu \left[\frac{\frac{1}{N} \sum_{n=0}^{N-1} \ell_\nu(\mathbf{s}_n; \mu(\mathbf{s}_n))}{\frac{1}{N} \sum_{n=0}^{N-1} \mathbb{I}(\mathbf{s}_n \in \mathcal{S}_{\text{comm}})} \right] = \frac{1}{\pi_{\text{comm}}} \int_{\mathcal{S}} \Pi_\mu(\mathbf{s}) \ell_\nu(\mathbf{s}; \mu(\mathbf{s})) d\mathbf{s}, \quad (18)$$

where $\Pi_\mu(\mathbf{s})$ is the steady-state probability density function of the SMDP being in a 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 in the SMDP, whose expression is provided in Appendix B, and $\ell_\nu(\mathbf{s}; \mathbf{a}) \triangleq$

$\Delta(s; \mathbf{a}) + \nu(E(s; \mathbf{a}) - P_{\text{avg}}T(s; \mathbf{a}))$ is the overall Lagrangian metric in state s under action \mathbf{a} . In (18), note that $\sum_{n=0}^{N-1} \ell_\nu(s_n; \mu(s_n))$ is the total Lagrangian cost accrued during the first N SMDP stages, and $\sum_{n=0}^{N-1} \mathbb{I}(s_n \in \mathcal{S}_{\text{comm}})$ is the number of communication states encountered in the SMDP; since a new decision interval is initiated after a communication state, this in turn equals the number of decision intervals. Therefore, after taking the expectation and the limit $N \rightarrow \infty$, $L_\mu^{(\nu)}$ represents the expected Lagrangian cost per decision interval, as expressed in (12). The right hand expression in (18) follows by noticing that the SMDP achieves a steady-state behavior when $N \rightarrow \infty$.

We now specialize the Lagrangian metric $\ell_\nu(s; \mathbf{a})$. Specifically, for waiting states,

$$\ell_\nu(r_U, \theta_U; v_r, \theta_c) = \nu \left(P_{\text{mob}} \left(\sqrt{v_r^2 + r_U^2 \theta_c^2} \right) - P_{\text{avg}} \right) \Delta_0; \quad (19)$$

for *communication* states under action $\xi = 0$, $\ell_\nu(r_U, \theta_U, r, \theta; 0, [\cdot]) = \frac{L}{R_{GB}(r)}$; and for *communication* states under action $\xi = 1$ with trajectory \mathbf{p}_U of duration Δ , we obtain

$$\ell_\nu(r_U, \theta_U, r, \theta; 1, \mathbf{p}_U) \triangleq (1 - \nu P_{\text{avg}}) \Delta + \nu \int_0^\Delta P_{\text{mob}}(v_U(\eta)) d\eta. \quad (20)$$

The minimization problem of (12) can then be expressed as the *average cost-per-stage problem*

$$g(\nu) = \frac{1}{\pi_{\text{comm}}} \min_{\mu} \int_{\mathcal{S}} \Pi_\mu(s) \ell_\nu(s; \mu(s)) ds, \quad (21)$$

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

Since GN transmission requests are uniformly distributed in the circular cell with the BS in the center, the UAV radius information is a sufficient statistic in decision making for a *waiting* state (r_U, θ_U) , which can be thus expressed as $s = r_U \in \mathcal{S}_{\text{wait}}$. 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 $s = (r_U, r, \psi) \in \mathcal{S}_{\text{comm}}$. A consequence of these sufficient statistics for decision making is that the policy affects the SMDP state transitions (hence steady-state behavior) only through the UAV radial velocity v_r in the *waiting* states and the UAV trajectory's target end radius position \hat{r}_U in the *communication* states. On the other hand, the angular velocity θ_c in the *waiting* states and the UAV trajectory's target end angular coordinate $\hat{\theta}_U$ in the *communication* states do not influence state dynamics, but only the instantaneous Lagrangian metric ℓ_ν .

With this observation, let $O(\mathbf{s}) \triangleq v_r \in [-V_{\max}, V_{\max}]$ define the radial velocity policy of the *waiting* states $\mathbf{s} \in \mathcal{S}_{\text{wait}}$, specifying the radial velocity component of a waiting action $\mathbf{a} = (v_r, \theta_c) \in \mathcal{A}_{\text{wait}}(\mathbf{s})$; let $U(\mathbf{s}) \triangleq \hat{r}_U \in [0, a]$ define the next radius position policy of the *communication* states $\mathbf{s} \in \mathcal{S}_{\text{comm}}$, specifying the end radius position of a scheduling and communication action $\mathbf{a} \in \mathcal{A}_{\text{comm}}(\mathbf{s})$. Under this decomposition, O and U constitute the outer decisions made by the SMDP and are the only actions to affect the steady-state distribution, denoted as $\Pi_{O,U}$ under the outer policy (O, U) . Thus, the optimization problem (21) can be restated as

$$g(\nu) = \frac{1}{\pi_{\text{comm}}} \min_{O,U} \left[\int_{\mathcal{S}_{\text{wait}}} \Pi_{O,U}(\mathbf{s}) \ell_{\nu}^*(\mathbf{s}; O(\mathbf{s})) d\mathbf{s} + \int_{\mathcal{S}_{\text{comm}}} \Pi_{O,U}(\mathbf{s}) \ell_{\nu}^*(\mathbf{s}; U(\mathbf{s})) d\mathbf{s} \right], \quad (22)$$

where ℓ_{ν}^* is the Lagrangian metric optimized with respect to the *inner action* components not specified by O and U . In particular, for waiting states $\mathbf{s} = r_U$ and radial velocity $O(\mathbf{s}) = v_r$, the inner optimization is performed with respect to the UAV angular velocity θ_c , i.e.,

$$\ell_{\nu}^*(\mathbf{s}; v_r) = \min_{\theta_c} \nu \left(P_{\text{mob}} \left(\sqrt{v_r^2 + r_U^2 \theta_c^2} \right) - P_{\text{avg}} \right) \Delta_0 \quad \text{s.t.} \quad \sqrt{v_r^2 + r_U^2 \theta_c^2} \leq V_{\max}. \quad (23)$$

Since $\nu \geq 0$, $\Delta_0 > 0$, and P_{avg} are constant, the optimizer θ_c^* is the angular velocity minimizing the UAV power consumption for a given UAV radial velocity v_r and radius r_U , solvable through exhaustive search. For communication states $\mathbf{s} = (r_U, r, \psi)$, $\ell_{\nu}^*(\mathbf{s}; \hat{r}_U)$ is determined by optimizing over the scheduling decision $\xi \in \{0, 1\}$ and, if $\xi = 1$, the trajectory \mathbf{p}_U followed by the UAV, terminating at the radial position \hat{r}_U . To formalize it, let $\ell_{\nu}^*(\mathbf{s}; \hat{r}_U, \xi)$ denote the optimized metric as a function of $\xi \in \{0, 1\}$. For $\xi = 1$ (D&F protocol),

$$\ell_{\nu}^*(\mathbf{s}; \hat{r}_U, 1) = \min_{\Delta, \mathbf{p}_U, t_p} (1 - \nu P_{\text{avg}}) \Delta + \nu \int_0^{\Delta} P_{\text{mob}} \left(\sqrt{r_U'(\eta)^2 + r_U^2(\eta) \theta_U'(\eta)^2} \right) d\eta \quad (24)$$

s.t. C.1, C.2,

$$\mathbf{p}_U(0) = (r_U, 0), \|\mathbf{p}_U(\Delta)\|_2 = \hat{r}_U, \quad (\hat{\text{C.3}})$$

where C.1 reflects the data payload constraints of the D&F protocol, C.2 is the maximum UAV velocity constraint, and $\hat{\text{C.3}}$ (rewritten version of C.3) enforces the trajectory constraints. For $\xi = 0$ (direct-BS, hence $\hat{r}_U = r_U$ ¹²), $\ell_{\nu}^*(\mathbf{s}; r_U, 0) = \frac{L}{R_{GB}(r)}$, hence $\ell_{\nu}^*(\mathbf{s}; r_U)$ is obtained by further

¹²Note that with direct transmission to the BS, the system immediately transitions to the waiting state r_U with the UAV at distance r_U from the center, hence $\hat{r}_U = r_U$ in this case.

minimizing over the scheduling decision $\xi \in \{0, 1\}$, yielding

$$\ell_\nu^*(\mathbf{s}; \hat{r}_U) = \min_{\xi \in \{0, 1\}} \ell_\nu^*(\mathbf{s}; r_U, \xi) \mathbb{I}(\hat{r}_U = r_U) + \ell_\nu^*(\mathbf{s}; \hat{r}_U, 1) \mathbb{I}(\hat{r}_U \neq r_U). \quad (25)$$

Thus, if the outer decision selects $U(\mathbf{s}) = r_U$, the inner scheduling decision $\xi \in \{0, 1\}$ is obtained by greedily minimizing a cost metric that trades off communication delay and energy consumption, i.e., direct transmission to the BS occurs if $\ell_\nu^*(\mathbf{s}; r_U, 0) < \ell_\nu^*(\mathbf{s}; r_U, 1)$. Otherwise, the UAV handles the GN request using the D&F protocol, and the inner decision on UAV trajectory greedily minimizes the instantaneous delay-power trade-off, terminating at the target radius of the outer decision, $U(\mathbf{s}) = \hat{r}_U$.

The optimization problem (24) will be the focus of Sec. IV. We now describe Algorithm 1, which designs the outer policy and computes the average cost-per-stage metric $g(\nu)$, along with the average energy- and time-per-stage metrics for a given ν , by solving problem (22) via value iteration. Here, we also discuss Algorithm 2, which solves the dual maximization problem $\max_{\nu \geq 0} g(\nu)$ via projected sub-gradient ascent [41].

Specifically¹³, in Algorithm 1, lines 2 and 3 compute the inner Lagrangian cost metric optimized with respect to the inner actions—along with the energy and time cost metrics—for all states and outer actions; line 6 computes the value iteration update for waiting states, which accounts for the transition to the waiting state $r_U + v_r \Delta_0$ (if no transmission request is received, w.p. $e^{-\Lambda \Delta_0}$) and to a communication state $(r_U + v_r \Delta_0, r', \psi')$ (if a transmission request is received in position (r', ψ') , with probability density function $\frac{1}{2\pi} \frac{2r'}{a^2}$); line 11 computes the value iteration update for communication states which—for end radius \hat{r}_U —accounts for the transition to the waiting state \hat{r}_U , w.p. 1; lines 7, 8, 12, and 13 update the corresponding total energy and time costs; line 17 estimates the values of the average cost-, energy-, and time-per-stage metrics.

In Algorithm 2, line 1 initializes the dual variable and a sequence of step-sizes used for projected dual sub-gradient ascent; line 3 calls the value iteration of Algorithm 1 using the current dual variable value ν_k , and outputs the optimal radial velocity (for waiting states) and next radius position (for communication states) policies, as well as the average cost-, energy-, and time-per-stage metrics; line 4 monitors convergence, checking the change in the average cost-per-stage value, as well as primal feasibility and complementary slackness, with respect to

¹³Although the pseudo-code in Algorithm 1 is defined on continuous state and action spaces, in a practical implementation, these spaces are discretized, so the integrals would be replaced by sums.

Algorithm 1 Value Iteration: $(O^*, U^*, g(\nu), \bar{E}, \bar{T}) = \text{VITER}(\nu)$

- 1: **Initialization:** $i=0$; the value function $V_i(\mathbf{s})=0$, total energy cost $E_i(\mathbf{s})=0$, and total time cost $T_i(\mathbf{s})=0$, for all states $\mathbf{s} \in \mathcal{S}$; stopping criterion δ .
 - 2: **Inner optimization in waiting states:** $\forall \mathbf{s} \in \mathcal{S}_{\text{wait}}, \forall v_r \in [-V_{\max}, V_{\max}]$, calculate $\ell_\nu^*(\mathbf{s}; v_r)$ as in (23), with minimizer θ_c^* ; compute energy cost $e^*(\mathbf{s}; v_r) = E(\mathbf{s}; v_r, \theta_c^*)$ and time cost $t^*(\mathbf{s}; v_r) = T(\mathbf{s}; v_r, \theta_c^*)$.
 - 3: **Inner optimization in communication states:** $\forall \mathbf{s} \in \mathcal{S}_{\text{comm}}, \forall \hat{r}_U \in [0, a]$, calculate $\ell_\nu^*(\mathbf{s}; \hat{r}_U)$ as in (24) and (25), with minimizer $(\xi^*, \hat{\theta}_U^*, \mathbf{p}_U^*, \Delta^*)$; compute energy cost $e^*(\mathbf{s}; \hat{r}_U) = \xi^* E(\mathbf{s}; \hat{r}_U, \hat{\theta}_U^*, \mathbf{p}_U^*)$ and time cost $t^*(\mathbf{s}; \hat{r}_U) = \xi^* T(\mathbf{s}; \hat{r}_U, \hat{\theta}_U^*, \mathbf{q}_U^*)$.
 - 4: **repeat**
 - 5: **for** each $\mathbf{s} = r_U \in \mathcal{S}_{\text{wait}}$ **do**
 - 6:
$$V_{i+1}(\mathbf{s}) \leftarrow \min_{v_r \in [-V_{\max}, V_{\max}]} \left[\ell_\nu^*(\mathbf{s}; v_r) + e^{-\Lambda \Delta_0} V_i(r_U + v_r \Delta_0) \right. \\ \left. + (1 - e^{-\Lambda \Delta_0}) \int_0^{2\pi} \frac{1}{2\pi} \int_0^a \frac{2r'}{a^2} V_i(r_U + v_r \Delta_0, r', \psi') dr' d\psi' \right],$$
 where $O_{i+1}(\mathbf{s})$ is the arg min.
 - 7:
$$E_{i+1}(\mathbf{s}) \leftarrow e^*(\mathbf{s}; O_{i+1}(\mathbf{s})) + e^{-\Lambda \Delta_0} E_i(r_U + O_{i+1}(\mathbf{s}) \Delta_0) \\ + (1 - e^{-\Lambda \Delta_0}) \int_0^{2\pi} \frac{1}{2\pi} \int_0^a \frac{2r'}{a^2} E_i(r_U + O_{i+1}(\mathbf{s}) \Delta_0, r', \psi') dr' d\psi'.$$
 - 8:
$$T_{i+1}(\mathbf{s}) \leftarrow t^*(\mathbf{s}; O_{i+1}(\mathbf{s})) + e^{-\Lambda \Delta_0} T_i(r_U + O_{i+1}(\mathbf{s}) \Delta_0) \\ + (1 - e^{-\Lambda \Delta_0}) \int_0^{2\pi} \frac{1}{2\pi} \int_0^a \frac{2r'}{a^2} T_i(r_U + O_{i+1}(\mathbf{s}) \Delta_0, r', \psi') dr' d\psi'.$$
 - 9: **end for**
 - 10: **for** each $\mathbf{s} = (r_U, r, \psi) \in \mathcal{S}_{\text{comm}}$ **do**
 - 11:
$$V_{i+1}(\mathbf{s}) \leftarrow \min_{\hat{r}_U \in [0, a]} [\ell_\nu^*(\mathbf{s}; \hat{r}_U) + V_i(\hat{r}_U)],$$
 where $U_{i+1}(\mathbf{s})$ is the arg min.
 - 12:
$$E_{i+1}(\mathbf{s}) \leftarrow e^*(\mathbf{s}; U_{i+1}(\mathbf{s})) + E_i(U_{i+1}(\mathbf{s}));$$
 - 13:
$$T_{i+1}(\mathbf{s}) \leftarrow t^*(\mathbf{s}; U_{i+1}(\mathbf{s})) + T_i(U_{i+1}(\mathbf{s})).$$
 - 14: **end for**
 - 15: $\forall \mathbf{s}$, calculate the stopping criterion metric $H(\mathbf{s}) = V_{i+1}(\mathbf{s}) - V_i(\mathbf{s})$; $i \leftarrow i+1$.
 - 16: **until** $\max_{\mathbf{s} \in \mathcal{S}} H(\mathbf{s}) - \min_{\mathbf{s} \in \mathcal{S}} H(\mathbf{s}) < \delta$.
 - 17: Approximate $[g(\nu); \bar{E}; \bar{T}] \approx \frac{1}{\pi_{\text{comm}}} \frac{1}{i} [V_i(\mathbf{s}); E_i(\mathbf{s}); T_i(\mathbf{s})]$, for some arbitrary $\mathbf{s} \in \mathcal{S}$.
 - 18: **return** $O^*(\mathbf{s}) = O_i(\mathbf{s}), \forall \mathbf{s} \in \mathcal{S}_{\text{wait}}, U^*(\mathbf{s}) = U_i(\mathbf{s}), \forall \mathbf{s} \in \mathcal{S}_{\text{comm}}, g(\nu), \bar{E}$, and \bar{T} .
-

the average UAV power constraint; line 7 updates the value of the dual variable in the direction of its sub-gradient and projects the value to the non-negative range to ensure dual feasibility. In the following section, we detail a hierarchical CSO algorithm for single-relay trajectory design (inner decision in *communication* states) within MAESTRO.

Algorithm 2 Projected Sub-gradient Ascent: PSGA()

```
1: Initialization:  $k=0$ ; dual variable  $\nu_0 \geq 0$ ; step-size  $\{\rho_j = \frac{\rho_0}{(j+1)}, j \geq 0\}$ ;  $g_{-1} = \infty$ .
2: for  $k=0, 1, \dots$  do
3:   Determine  $(O_k^*, U_k^*, g_k, \bar{E}_k, \bar{T}_k) = \text{VITER}(\nu_k)$  via Algorithm 1.
4:   if  $|g_k - g_{k-1}| < \epsilon_{DI}$ ;  $\bar{E}_k - P_{\text{avg}} \bar{T}_k < \epsilon_{PF}$ ;  $\nu_k |\bar{E}_k - P_{\text{avg}} \bar{T}_k| < \epsilon_{CS}$  then
5:     return: optimal outer policy  $(O_k^*, U_k^*)$ ;
6:   else
7:     Update  $\nu_{k+1} = \max \{ \nu_k + \rho_k (\bar{E}_k - P_{\text{avg}} \bar{T}_k), 0 \}$ ;  $k \leftarrow k+1$ .
8:   end if
9: end for
```

IV. MAESTRO: TRAJECTORY DESIGN VIA HIERARCHICAL CSO

In this section, we design the UAV trajectory during the D&F protocol by solving (24). The literature solves problems like (24) with SCA (see [12], [13], [42]), in which the non-convex objective and constraints are approximated by convex functions around a local point; the approximated problem is solved with convex optimization methods, the minimizers become the next local point, convex approximations are made about the new point, and the process repeats until convergence. However, SCA has major drawbacks: 1) it is computationally slow, as shown numerically in Sec. VI; 2) its QoS constraints on the GN data payload require additional approximations, which are not employed in our model, and these preclude the adoption of rate adaptation at the transmitters to efficiently exploit the A2G channel stochastics.

Within MAESTRO, to overcome these challenges, we propose a framework based on CSO [27]. Since the design space involves optimizing over infinitely many variables $(r_U(t), \theta_U(t))$, 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 suffers from poor convergence (see Fig. 5a in Sec. VI), because solution spaces of such problems tend to increase exponentially with dimensionality [43]. We address this weakness by proposing a Hierarchical-CSO (HCSO) algorithm, in which a sequence of CSO problems is solved: initially, CSO produces a low-resolution trajectory; the optimized trajectory is then used to create a higher-resolution trajectory via interpolation, which is further optimized with CSO; and so on. This sequence of CSO problems generate trajectories of increasing resolution and helps alleviate the poor performance achieved when optimizing the high-resolution trajectory directly, as demonstrated in Fig. 5a.

First, we describe the meta-heuristic UAV trajectory to be solved with HCSO and convert the constrained problem into an unconstrained one, so as to make the problem tractable for the CSO algorithm; next, we design the hierarchical structure within HCSO, and specifically show how high-dimensional trajectories leverage information from low-dimensional ones to improve the solution accuracy.

Formulation of the Meta-Heuristic UAV Trajectory: To define the UAV meta-heuristic trajectory, we use Cartesian coordinates instead of polar coordinates, for easier evaluation of the data payload constraints. For a state $\mathbf{s}=(r_U, r, \psi) \in \mathcal{S}_{\text{comm}}$ and a next radius position $U(\mathbf{s})=\hat{r}_U$, we encode a trajectory as a sequence of way-points and UAV velocities. In particular, we define a meta-heuristic trajectory of \mathbf{p}_U by a sequence of M trajectory segments, $\{\mathbf{x}_m\}_{m=0}^M=\{(x_m, y_m)\}_{m=0}^M$. The UAV trajectory begins at \mathbf{x}_0 and ends at \mathbf{x}_M ; the UAV traverses each straight trajectory segment $\Psi_m \triangleq \mathbf{x}_{m+1}-\mathbf{x}_m$ with velocity $v_m \in [V_{\text{low}}, V_{\text{max}}]$, with minimum velocity $V_{\text{low}} \ll V_{\text{max}}$ introduced to ensure well-defined durations; the sequences of trajectory way-points $\mathbf{p} \triangleq [\mathbf{x}_1, \dots, \mathbf{x}_M]^T$ and velocities $\mathbf{v} \triangleq [v_0, \dots, v_{M-1}]^T$ are the optimization variables. For M trajectory segments, the first and second $\frac{M}{2}$ segments correspond to the two-phase D&F protocol. As the throughput equations are angular-invariant, without loss of generality, the initial UAV location is set to $\mathbf{x}_0=(r_U, 0)$ and the GN request location to $\mathbf{x}_G \triangleq (r \cos \psi, r \sin \psi)$. Since the number of bits communicated (C.1) during each trajectory segment cannot be computed in closed-form, we approximate them for each trajectory segment numerically. Specifically, between subsequent way-points, \mathbf{x}_m and \mathbf{x}_{m+1} with velocity v_m , we generate a sequence of n_{res} evenly-spaced points with a sufficiently high resolution, i.e., $\{\mathbf{x}_k^{\text{new}}\}_{k=0}^{n_{\text{res}}}$; the time durations between the sequence of points, $\{t_k^{\text{new}}\}_{k=0}^{n_{\text{res}}-1}$, and expected throughput (corresponding to D&F protocol) at each point, $\{R_k^{\text{new}}\}_{k=0}^{n_{\text{res}}-1}$, then determine the number of bits communicated for the m th trajectory segment by calculating the lower Riemann sum, given by $F_m \triangleq \sum_{k=0}^{n_{\text{res}}-1} t_k^{\text{new}} R_k^{\text{new}}$. With the number of communicated bits evaluated for each trajectory segment, we present the heuristic problem of

$\ell_\nu^*(\mathbf{s}; U(\mathbf{s}))$ to be solved with HCSO. The problem in standard form is

$$(\mathbf{P}.0) \quad \min_{\mathbf{p} \in \mathbf{v}} (1 - \nu P_{\text{avg}}) \sum_{m=0}^{M-1} \frac{\|\Psi_m\|_2}{v_m} + \nu \sum_{m=0}^{M-1} \frac{\|\Psi_m\|_2}{v_m} P_{\text{mob}}(v_m) \quad (26)$$

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

$$\mathbf{x}_0 = (r_U, 0), \quad \mathbf{x}_M = \hat{r}_U \frac{\mathbf{x}_{M-1}}{\|\mathbf{x}_{M-1}\|_2} \mathbb{I}(\|\mathbf{x}_{M-1}\|_2 \neq 0) + (\hat{r}_U, 0) \mathbb{I}(\|\mathbf{x}_{M-1}\|_2 = 0), \quad (\tilde{\mathbf{C}}.3)$$

where $\tilde{\mathbf{C}}.1$ (rewritten version of C.1) enforces the data payload constraint and $\tilde{\mathbf{C}}.3$ (rewritten version of C.3) enforces the UAV trajectory constraints. Particularly, the end radius constraint is satisfied by projecting the way-point \mathbf{x}_{M-1} to the nearest point with radius \hat{r}_U , and $\frac{\|\Psi_m\|_2}{v_m}$ is the duration of the m th trajectory segment. Thus, the final way-point variable \mathbf{x}_M is eliminated, and \mathbf{x}_0 is fixed.

We solve (P.0) with HCSO, as it promises to provide results efficiently. The HCSO method is executed by converting the constrained problem (P.0) into an unconstrained one and solving a sequence of unconstrained problems with CSO. We remove the data payload constraints by adding them to the objective function as penalties for constraint violations by enforcing a particular solution: if the UAV does not *decode* (or *forward*) its data payload at 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 ($v_{P_{\min}} = 22 \text{ m s}^{-1}$ [12]) until it receives (or transmits) this data payload. This penalized objective function is

$$\begin{aligned} \hat{f}(\mathbf{p}, \mathbf{v}) \triangleq & (1 - \nu P_{\text{avg}}) \sum_{m=0}^{M-1} \frac{\|\Psi_m\|_2}{v_m} + \nu \sum_{m=0}^{M-1} \frac{\|\Psi_m\|_2}{v_m} P_{\text{mob}}(v_m) \\ & + (1 - \nu P_{\text{avg}})(\hat{t}_{P,1} + \hat{t}_{P,2}) + \nu(\hat{E}_{P,1} + \hat{E}_{P,2}), \end{aligned} \quad (27)$$

where the time delay penalties in the first and second phases, respectively, are

$$\hat{t}_{P,1} \triangleq \frac{h_1(\mathbf{p}, \mathbf{v})}{\bar{R}_{GU}(\|\mathbf{x}_{M/2} - \mathbf{x}_G\|_2)} \mathbb{I}(h_1(\mathbf{p}, \mathbf{v}) > 0), \text{ and} \quad (28)$$

$$\hat{t}_{P,2} \triangleq \frac{h_2(\mathbf{p}, \mathbf{v})}{\bar{R}_{UB}(\|\mathbf{x}_M\|_2)} \mathbb{I}(h_2(\mathbf{p}, \mathbf{v}) > 0); \quad (29)$$

similarly, the energy penalties in the first and second phases, respectively, are $[\hat{E}_{P,1}; \hat{E}_{P,2}] \triangleq P_{\text{mob}}(v_{P_{\min}})[\hat{t}_{P,1}; \hat{t}_{P,2}]$.

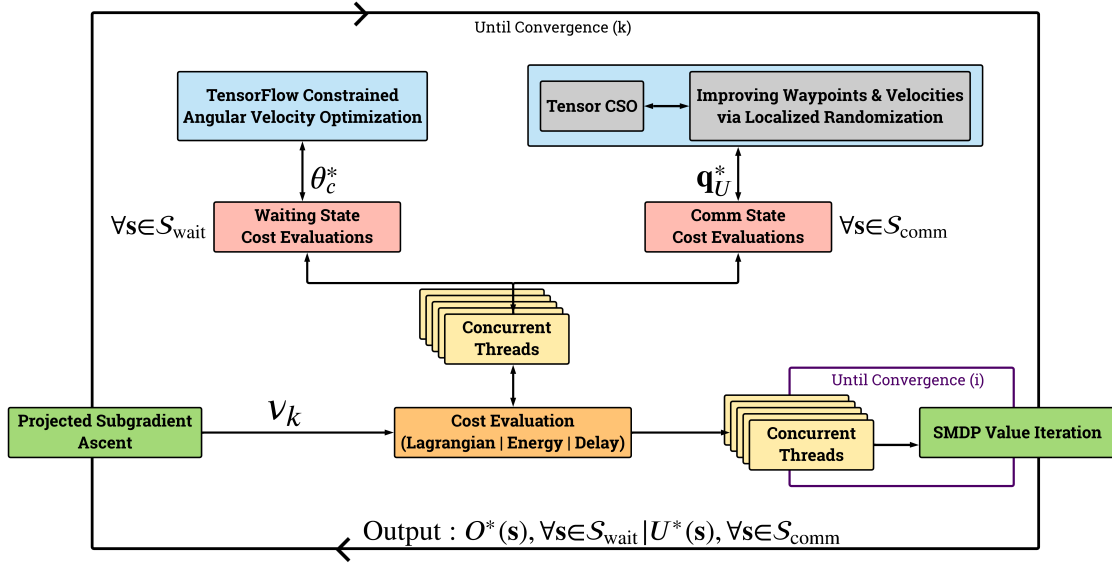


Fig. 3: A depiction of the algorithmic flow inherent in our single-relay policy optimization process; computing accelerations via multi-threaded concurrency and distributed computing are also illustrated here.

A compact illustration of the sequence of operations involved in MAESTRO to solve for the optimal *waiting* and *communication* state policies—via Algorithms 1, 2, 3, and 4—is depicted in Fig. 3.

Solution of (P.0) via the HCSO Method: To solve the problem with HCSO, we begin by randomly initializing a population of M -segment trajectory solutions (particles). Specifically, in the first call to the CSO algorithm [27] (see Algorithm 4) inside the HCSO framework, we initialize N trajectory particles $\mathbf{p}_{1:N} \triangleq \mathbf{p}_1, \dots, \mathbf{p}_N$, with particle velocities $\mathbf{u}_{1:N} \triangleq \mathbf{u}_1, \dots, \mathbf{u}_N$, and N UAV velocity particles $\mathbf{v}_{1:N} \triangleq \mathbf{v}_1, \dots, \mathbf{v}_N$, with particle velocities $\mathbf{w}_{1:N} \triangleq \mathbf{w}_1, \dots, \mathbf{w}_N$, imposing no trajectory structure in the first iteration. After the first CSO iteration, the resulting trajectory is interpolated to form a reference $2M$ -segment trajectory, $\tilde{\mathbf{p}} = [\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_{M-1}]^T$ and $\tilde{\mathbf{v}} = [\tilde{v}_0, \dots, \tilde{v}_{M-1}]^T$, with $M \leftarrow 2M$, so as to increase the resolution for the next iteration. The new population size is then reduced, $N \leftarrow N - N_{\text{red}}$, to lower the computational burden of CSO, and a new set of trajectories is generated randomly in a neighborhood around the reference trajectory. More precisely, the m th way-point, $\mathbf{x}_m = (x_m, y_m)$, of a new trajectory is generated as $\mathbf{x}_m = \tilde{\mathbf{x}}_m + (\chi_m, \zeta_m)$, where the localized randomness in the new way-points is modeled using zero-mean Gaussian random variables, $\chi_m, \zeta_m \sim \mathcal{N}(0, \varsigma (\|\tilde{\mathbf{x}}_{m+1} - \tilde{\mathbf{x}}_m\|^2 + \|\tilde{\mathbf{x}}_{m-1} - \tilde{\mathbf{x}}_m\|^2))$, with $\varsigma > 0$ as a scaling factor, and the variance determined by the spread between neighboring reference trajectory way-points, incorporated due to the empirical observation that areas with clustered UAV way-points are the regions where the objective function (27) is sensitive to

Algorithm 3 Hierarchical Competitive Swarm Optimization: HCSO(s; $U(s)$)

```
1: Initialization: Randomly initialize UAV way-points  $\mathbf{p}_{1:N}$  and velocities  $\mathbf{v}_{1:N}$ .
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 Algorithm 4).
4:   Increase  $M \leftarrow 2M$ ; interpolate to form reference trajectory:  $(\tilde{\mathbf{p}}, \tilde{\mathbf{v}}) = \text{interp}(\mathbf{p}^*, \mathbf{v}^*, M)$ .
5:   Reduce population size  $N \leftarrow N - N_{\text{red}}$ .
6:   for  $n_1 = 1, 2, \dots, N$  do
7:     for  $n_2 = 1, 2, \dots, M-1$  do
8:        $\mathbf{x}_{n_2} = \tilde{\mathbf{x}}_{n_2} + (\chi_{n_2}, \zeta_{n_2})$  with  $\tilde{\mathbf{x}}_{n_2} \in (\tilde{x}_{n_2}, \tilde{y}_{n_2})$  in  $\tilde{\mathbf{p}}$ ;  $\mathbf{x}_{n_2} \in \mathbf{p}_{n_1}$ .
9:     end for
10:    for  $n_3 = 0, \dots, M-1$  do
11:       $\tilde{v}_{n_3} = [\tilde{v}_{n_3} + \varkappa_{n_3}]^{[V_{\text{low}}, V_{\text{max}}]}$  with  $\tilde{v}_{n_3} \in \tilde{\mathbf{v}}$ ;  $v_{n_3} \in \mathbf{v}_{n_1}$ .
12:    end for
13:    Update new swarm particle  $\mathbf{p}_{n_1} \leftarrow [\mathbf{x}_1, \dots, \mathbf{x}_{M-1}]^T$  and  $\mathbf{v}_{n_1} \leftarrow [v_0, \dots, v_{M-1}]^T$ .
14:  end for
15: end while
```

large variations. Likewise, the UAV velocity of the m th new trajectory segment is generated by $v_m = [\tilde{v}_m + \varkappa_m]^{[V_{\text{low}}, V_{\text{max}}]}$ where $\varkappa_m \sim \mathcal{N}(0, \varepsilon (V_{\text{max}} - V_{\text{low}})^2)$ induces randomness in the new velocity, $\varepsilon > 0$ is a scaling factor, and $[\cdot]^{[V_{\text{low}}, V_{\text{max}}]} = \max\{\min\{\cdot, V_{\text{max}}\}, V_{\text{low}}\}$ projects velocity v_m to the feasible set. The variance is chosen due to the observation that the UAV velocities exhibit consistent convergence with CSO much faster than the trajectory way-points and are less sensitive to random initializations.

The HCSO scheme is outlined in Algorithm 3. Line 1 initializes N random M -segment UAV trajectories; line 3 obtains an M -segment trajectory with CSO [27] (Algorithm 4); line 4 doubles the trajectory resolution and interpolates the CSO output to a $2M$ -segment reference trajectory; lines 8 and 11 generate random trajectories in a neighborhood of the reference trajectory, $(\tilde{\mathbf{p}}, \tilde{\mathbf{v}})$. The process repeats until the trajectory resolution equals a desired value.

The CSO scheme follows [27] and is shown in Algorithm 4. During the k th iteration inside CSO, the N particles are randomly grouped into $\frac{N}{2}$ pairwise competitions. For both members of a pair, the penalized objective function in (27) is calculated; the winner of the competition is passed onto the $(k+1)$ th iteration of CSO, and the loser is modified by learning from the winner. Let

Algorithm 4 Competitive Swarm Optimization: $(\mathbf{p}^*, \mathbf{v}^*) = \text{CSO}(\mathbf{p}_{1:N}(0), \mathbf{v}_{1:N}(0), N, M)$

```
1: Initialization:  $k=0$ , particle velocities  $\mathbf{u}_{1:N}(0)$  and  $\mathbf{w}_{1:N}(0)$  for  $\mathbf{p}_{1:N}(0)$  and  $\mathbf{v}_{1:N}(0)$ .
2: repeat
3:   Shuffle particle indices with a random permutation:  $t(i) : \{1, \dots, N\} \mapsto \{1, \dots, N\}$ .
4:   for  $j=1, 3, 5, \dots, N-1$  do
5:     if  $\hat{f}(\mathbf{p}_{t(j)}(k), \mathbf{v}_{t(j)}(k)) \leq \hat{f}(\mathbf{p}_{t(j+1)}(k), \mathbf{v}_{t(j+1)}(k))$  then
6:        $(\mathbf{p}^w, \mathbf{v}^w) = (\mathbf{p}_{t(j)}(k), \mathbf{v}_{t(j)}(k)); (\mathbf{p}^l, \mathbf{v}^l) = (\mathbf{p}_{t(j+1)}(k), \mathbf{v}_{t(j+1)}(k))$ 
7:        $j_{\text{win}} = t(j); j_{\text{los}} = t(j+1)$ .
8:     else
9:        $(\mathbf{p}^w, \mathbf{v}^w) = (\mathbf{p}_{t(j+1)}(k), \mathbf{v}_{t(j+1)}(k)); (\mathbf{p}^l, \mathbf{v}^l) = (\mathbf{p}_{t(j)}(k), \mathbf{v}_{t(j)}(k))$ 
10:       $j_{\text{win}} = t(j+1); j_{\text{los}} = t(j)$ .
11:    end if
12:    Pass the winner to next iteration:  $(\mathbf{p}_{j_{\text{win}}}(k+1), \mathbf{v}_{j_{\text{win}}}(k+1)) \leftarrow (\mathbf{p}^w, \mathbf{v}^w)$ .
13:    Update loser particle velocities  $(\mathbf{u}_{j_{\text{los}}}(k+1), \mathbf{w}_{j_{\text{los}}}(k+1))$  using (30) and (31).
14:    Update loser particles  $(\mathbf{p}_{j_{\text{los}}}(k+1), \mathbf{v}_{j_{\text{los}}}(k+1))$  using (32).
15:  end for
16:   $k \leftarrow k+1$ .
17: until maximum number of cost function evaluations performed.
    return  $(\mathbf{p}^*, \mathbf{v}^*) = \arg \min \{ \hat{f}(\mathbf{p}_i(k), \mathbf{v}_i(k)), \forall i=1, \dots, N \}$ 
```

$j_{\text{win}}, j_{\text{los}} \in \{1, \dots, N\}$ denote the indices of the winner and the loser from the population during the j th pairwise competition, in the k th iteration of CSO. Accordingly, we denote the winner and loser particles by $(\mathbf{p}^w, \mathbf{v}^w)$ and $(\mathbf{p}^l, \mathbf{v}^l)$, respectively. The loser particle and its velocities are updated as

$$\mathbf{u}_{j_{\text{los}}}(k+1) = r_{j,1}(k)\mathbf{u}^l + r_{j,2}(k)(\mathbf{p}^w - \mathbf{p}^l) + \omega \cdot r_{j,3}(k)(\bar{\mathbf{p}}(k) - \mathbf{p}^l), \quad (30)$$

$$\mathbf{w}_{j_{\text{los}}}(k+1) = r_{j,1}(k)\mathbf{w}^l + r_{j,2}(k)(\mathbf{v}^w - \mathbf{v}^l) + \omega \cdot r_{j,3}(k)(\bar{\mathbf{v}}(k) - \mathbf{v}^l), \text{ and} \quad (31)$$

$$\mathbf{p}_{j_{\text{los}}}(k+1) = \mathbf{p}^l + \mathbf{u}_{j_{\text{los}}}(k+1), \quad \mathbf{v}_{j_{\text{los}}}(k+1) = [\mathbf{v}^l + \mathbf{w}_{j_{\text{los}}}(k+1)]^{[V_{\text{low}}, V_{\text{max}}]}, \quad (32)$$

where $r_{j,1}(k), r_{j,2}(k)$, and $r_{j,3}(k)$ are independent uniform random variables on the interval $[0, 1]$; $\bar{\mathbf{p}}(k)$ and $\bar{\mathbf{v}}(k)$ are the global means of the UAV way-point and velocity particles in the k th CSO iteration; and ω is a scale parameter for the degree of influence of $\bar{\mathbf{p}}(k)$ and $\bar{\mathbf{v}}(k)$.

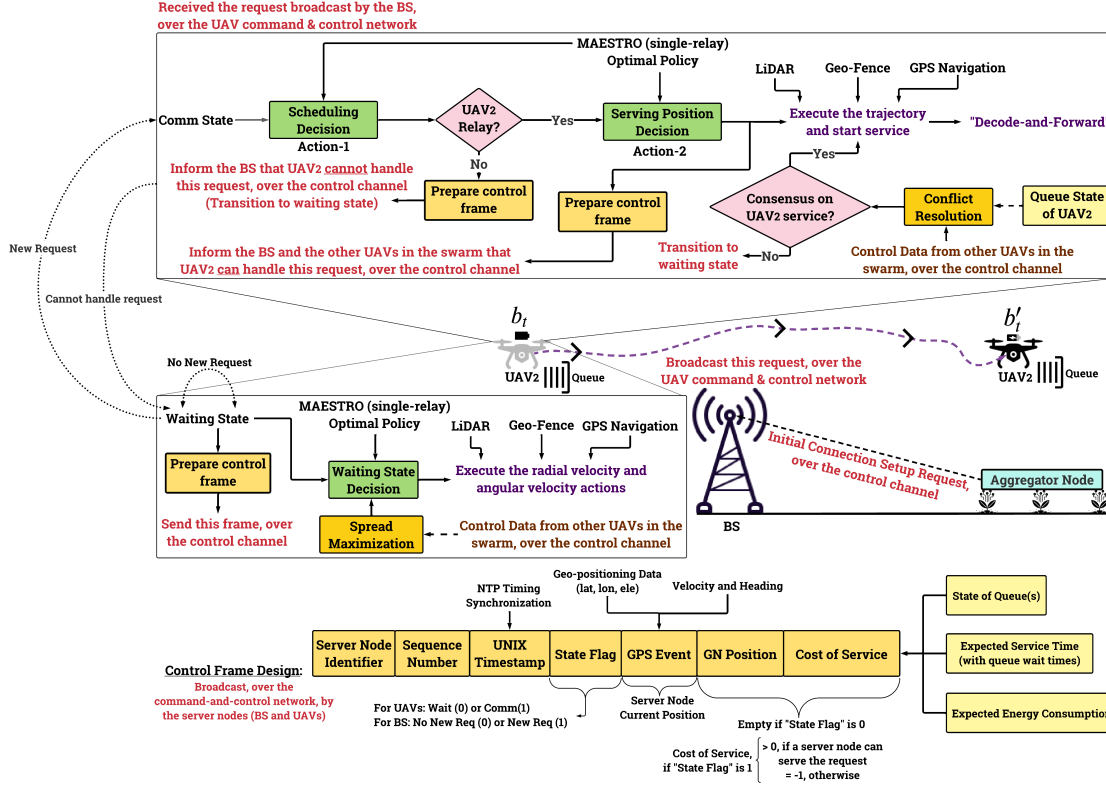


Fig. 4: The study of a hybrid wireless network augmented by a swarm of UAV relays in precision agriculture: an illustration outlining the sequence of operations under MAESTRO-X that occur at UAV₂ in the swarm; our control frame structure is also depicted here.

Computing Accelerations: In order to further boost policy training times, and turn our proposed scheme into a scalable and tractable solution, we incorporate efficient design methodologies for the implementation of Algorithms 1, 2, 3, and 4 in software. Leveraging a fully-vectorized implementation in TensorFlow [44], we perform concurrent executions (via Python worker threads) of appropriately disassociated program blocks. These concurrent executions exploit distributed computing capabilities available at our disposal via context-based device placement over CUDA-capable GPUs and just-in-time accelerated CPUs [45]. The computing accelerations outlined above facilitate significant improvements in policy convergence times, as discussed in Sec. VI. A multi-relay extension to MAESTRO is described next.

V. MAESTRO-X: AN EXTENSION TO UAV SWARMS VIA SUPPLEMENTARY HEURISTICS

In this section, we remove the specializations considered for our single-relay construction in Sec. III, and extend the optimal scheduling and trajectory optimization framework (MAESTRO) developed in Secs. III and IV to our generalized hybrid wireless network model of N_U UAV relays in the swarm. MAESTRO's eXtension, termed MAESTRO-X, constitutes augmentations

to the multiscale optimal *waiting* and *communication* policy obtained via SMDP value iteration (within which HCSO determines the service trajectory). Depicting an example scenario of serving data traffic generated by an aggregation of soil sensors in precision agriculture, Fig. 4 illustrates the control flow of MAESTRO-X—in particular, the operations of UAV₂. Along with a fully-connected mesh network overlaid over the BS and the UAV swarm for command-and-control operations, enhancements to MAESTRO include spread maximization (in the *waiting* phase) and conflict resolution (in the *request scheduling* phase). After these heuristics are embedded into MAESTRO, the modified policy is replicated across the swarm, resulting in an implementation centered around multiple decoupled executions driven by a single optimal policy.

Policy Replication: Post policy convergence via the algorithms detailed in Secs. III and IV, the optimal behavior of our multiscale adaptive framework developed for the automation of single-relay operations (i.e., MAESTRO) is stored in a database: an organized collection of key-value pairs. Staying true to the SMDP state space construction, this database is partitioned into two lookup tables. The *waiting* phase lookup table constitutes the SMDP waiting states as the keys and their associated radial and angular velocity actions as the corresponding values. Similarly, the *request scheduling* phase lookup table constitutes the SMDP communication states as the keys and their associated UAV service trajectories and service times (for both the BS and the UAVs) as the corresponding values. This optimal policy database is then replicated to the BS as well as all the UAVs in the swarm. During their operations, these server nodes (BS + UAVs) lookup the optimal actions vis-à-vis their current states—and, with further enhancements such as collaboration message exchanges, spread maximization (in the *waiting* phase) and consensus-driven conflict resolution (in the *request scheduling* phase)—execute optimal actions that have been augmented to orchestrate swarm operations in a decentralized fashion.

Command-and-Control Network: Truly decentralized and coordinated operations of the UAV relays in the swarm necessitates the need for a control network over which the server nodes can collaborate to 1) ensure collision-free movements among the UAVs¹⁴; 2) facilitate consensus-driven system-wide decisions on the best server to handle a GN uplink transmission request that originated in the cell, thereby mitigating the additional computational complexity endured by joint/combined multi-relay optimization frameworks [13], [14]; and 3) guarantee resilient fault-

¹⁴We reasonably assume that the UAVs are equipped with a LiDAR-based obstacle avoidance system: when an obstacle is detected during the UAV's flight, a fallback guidance system on-board navigates the UAV around the obstacle. The incorporation of obstacles and No-Fly-Zones into our model will be tackled in future works.

tolerant operations by setting-up fallback mechanisms to handle UAV-failures. We designate the band-edges of the allocated spectrum as control channels, over which the server nodes in the cell exchange short frames: the structure of a control frame in MAESTRO-X is shown in Fig. 4. A fully-connected distributed mesh topology (employing these designated control channels) overlaid over the BS and the UAV swarm constitutes the design of our command-and-control network. Since each UAV relay in the swarm possesses the same optimal *waiting* and *communication* state policies, we introduce spread maximization (in the *waiting* states) and conflict resolution heuristics (in the *communication* states) to cooperatively handle their operations. These heuristics are discussed next.

Spread Maximization: To efficiently position and prime the idle UAVs for a potential new GN request originating in the cell, in the *waiting* states, a UAV relay in the swarm, in addition to executing the optimal action (corresponding to its current radius level), determines the direction of angular motion (clockwise or counter-clockwise) based on our spread maximization heuristic—wherein each UAV relay in the *waiting* state and in a radius level with zero radial velocity, executes either positive (counter-clockwise) or negative (clockwise) angular movements in order to maximize the minimum distance among them. These coordinated movements among the UAVs is made possible through periodic exchanges of control frames over the command-and-control network. Studying this frame structure in Fig. 4, we note that for waiting states, ‘State Flag’ is set to 0, and the ‘GN Position’ and ‘Cost of Service’ fields are empty. The ‘GPS Event’ field constitutes the most crucial component in a *waiting* state control frame: UAVs in the *waiting* state extract the positional information of their peers from this field, and apply a *maximize-the-minimum-distance* heuristic over other UAVs which are *waiting* in the same radius level. This methodology ensures that a suitable spread is maintained among the idle UAVs, in order to facilitate faster response times when a new uplink request is generated.

Consensus-driven Conflict Resolution: When a new uplink transmission request originates in the cell, the UAVs already serving a GN continue to do so, i.e., they do not participate in the consensus-driven conflict resolution process. These relays, termed ‘unavailable’ under this context, transition into their corresponding *waiting* states upon service completion. On the other hand, UAVs in the *waiting* states transition into their respective *communication* states. The BS along with these relays are deemed to be ‘available’. These ‘available’ server nodes exchange collaboration messages over the command-and-control network. The structure of this control frame is depicted in Fig. 4: ‘State Flag’ is set to 1, the ‘GN position’ field is populated with

the originating position of the uplink transmission request under consideration, and the ‘Cost of Service’ field constitutes a weighted sum of the request service time (communication service + queue wait) and the necessary energy consumption to serve it. These values on communication service times and energy consumption are obtained from the policy lookup table (whose design was discussed earlier in this section); the queueing latencies at these server nodes are obtained by modeling an $M/G/N_B$ queueing system at the BS and an $M/G/1$ queueing system at each UAV relay. Note that the BS’ energy consumption for serving a GN request is set to 0. Upon sharing these metrics with each other, these nodes arrive at a consensus on the most efficient choice (vis-à-vis service delay and energy consumption) to successfully handle the active GN request. Next, we chronicle our simulation setup and the corresponding results from our numerical evaluations of both MAESTRO and MAESTRO-X.

VI. THE SIMULATION SETUP AND NUMERICAL EVALUATIONS

As outlined in Table II, in our numerical evaluations of MAESTRO and MAESTRO-X, we use a channel bandwidth of $B = 5$ MHz; for all links, NLoS attenuation constant $\kappa = 0.2$, 1-meter reference SNR $\frac{\beta_0 P}{\sigma^2 \Gamma} = 40$ dB, LoS path-loss exponent $\alpha = 2$, NLoS path-loss exponent $\tilde{\alpha} = 2.8$, Rician K -factor parameters $k_1 = 1$ and $k_2 = 0.05$ [26], and LoS probability parameters $z_1 = 9.61$ and $z_2 = 0.16$ [46]; UAV height $H_U = 200$ m; BS antenna height $H_B = 80$ m; maximum UAV velocity $V_{\max} = 55 \text{ m s}^{-1}$; and cell radius $a = 1000$ m. The UAV power consumption model uses the relationship described in (5) and the parameters listed Table II (according to the studies detailed in [12]). We solve an approximation of problem (22) by discretizing the state and action spaces and applying linearly-interpolated value iteration. The dual variable yields meaningful results for $\nu \in [0, \frac{1}{P_{\text{avg}}}]$, thus by solving the problem for 20 values in the interval, an approximation to the maximum $g(\nu^*)$, emerges. We discretize the states with $N_{\text{sp}} = 25$ equispaced radii values, with 12 GNs in each level; and $R_{\text{sp}} = 25$ equispaced radial velocity actions for $v_r \in \{-V_{\max}, \dots, V_{\max}\}$. Lastly, Δ_0 is chosen to satisfy $e^{-\Lambda \Delta_0} = 0.93$, so that it is unlikely to receive two or more requests in Δ_0 .

[BK: In progress: To benchmark HCSO against SCA and non-hierarchical CSO for solving the inner Lagrangian cost metric, $\ell_\nu^*(s; U(s))$, we choose state $s = (800, 500, \frac{\pi}{4})$ [m,m,rad], end radius $U(s) = 700$ m, target average power $P_{\text{avg}} = 1.2$ kW, and data payload 10 Mb. For HCSO (Algorithm 3), with the initial trajectory size $M = 16$ and the maximum allowed trajectory size $M_{\max} = 256$, we make 4 calls to CSO, doubling the trajectory resolution each

Notation(s)	Physical Meaning/Description	Simulation Value(s)
a	Radius of the cell	1000 m
(x_B, y_B)	Rectangular coordinates of the BS	(0 m, 0 m)
H_B	GPS ellipsoidal height of the BS antennae	80 m
H_U	GPS ellipsoidal height of the antennae on-board the UAV Relay(s)	200 m
N_G	Number of GNs in the cell	300
N_U	Number of UAV relays in the formulation	[1, 2, 3]
λ_G	GN distribution density in the cell	9.55×10^{-6} GNs/m ²
L	Data payload size to be serviced for each GN request	[1, 10, 100] Mb
$\lambda_{R G}$	Rate of the Poisson arrival process for requests originating at each GN	$[5.6 \times 10^{-4}, 1.1 \times 10^{-4}, 1.9 \times 10^{-5}]$ req/GN/s
λ_R	Rate of the Poisson arrival process for requests per unit area	$[5.3 \times 10^{-9}, 1.1 \times 10^{-9}, 1.8 \times 10^{-10}]$ req/s/m ²
Λ	Overall rate of requests originating in the cell	$[1.7 \times 10^{-2}, 3.3 \times 10^{-3}, 5.6 \times 10^{-4}]$ req/s
Δ_0	UAV relay(s) waiting-state action execution time-frame	[4.35, 21.79, 130.52] s
U_{tip}	Rotor blade tip velocity in the rotary-wing UAV	200 m s ⁻¹
v_0	Mean rotor-induced velocity in the rotary-wing UAV	7.2 m s ⁻¹
v_{max}	Maximum forward flying velocity constraint value for the UAV relay(s)	55 m s ⁻¹
P_1	UAV mobility power profile constant vis-à-vis the blade profile	580.65 W
P_2	UAV mobility power profile constant vis-à-vis the induced velocity	790.6715 W
P_3	UAV mobility power profile constant vis-à-vis parasite terms	0.0073 kg m ⁻¹
P_{avg}	Average mobility power constraint value for the UAV relay(s)	$\{1.0 + j(0.1), \forall j \in \{0, 1, 2, \dots, 10\}\}$ W
B	Bandwidth of each channel used for GB, GU, and UB links	5 MHz
$\gamma_{\text{GB}} = \gamma_{\text{GU}} = \gamma_{\text{UB}}$	The SNR at a reference Tx-Rx distance of 1 m	40 dB
α	Path-loss exponent for LoS links	2
k_1	Radio environment coefficient of the Rician LoS K -factor	1
k_2	Radio environment coefficient of the Rician LoS K -factor	0.051
$\tilde{\alpha}$	Path-loss exponent for NLoS links	2.8
κ	Additional NLoS attenuation factor	0.2
z_1	Radio environment coefficient for LoS/NLoS probability evaluation	9.61
z_2	Radio environment coefficient for LoS/NLoS probability evaluation	0.16
R_{sp}	Number of UAV radial velocity levels	25
N_{sp}	Number of radius levels in the cell	25
ϵ_Z	Bisection method tolerance for rate adaptation	1×10^{-10}
α_{θ_c}	Learning rate for UAV angular velocity optimization	1×10^{-4}
ϵ_{θ_c}	Termination threshold for UAV angular velocity optimization	1×10^{-10}
δ	Termination threshold for SMDP Value Iteration	1×10^{-3}
ϵ_{PF}	Primal feasibility threshold in projected sub-gradient ascent	1×10^{-3}
ρ_0	Initial dual step-size in projected sub-gradient ascent	1
ϵ_{DI}	Dual convergence threshold in projected sub-gradient ascent	1×10^{-3}
ϵ_{CS}	Complementary slackness threshold in projected sub-gradient ascent	0.1
N_{max}	Maximum number of cost function evaluations in CSO	1000
M	Initial number of trajectory segments in HCSO	16
M_{max}	Maximum number of trajectory segments in HCSO	256
N	Initial number of trajectory and velocity particles in HCSO	400
$\zeta = \epsilon = \omega$	Scale parameters in HCSO	1

TABLE II: Simulation Parameters

iteration as $M_{\text{HCSO}}(i) = 2^{i+1}$, for $i = 1, \dots, 4$; we decrease the swarm size after each CSO call as $N_{\text{HCSO}}(i) = 400 - 20i$, for $i = 1, \dots, 4$. In Fig. 5a, we observe that the HCSO realizations converge up to $21 \times$ faster in time. Thus, HCSO is preferable to SCA for time efficiency, considering that $\approx 10,000$ realizations must be solved with the SMDP discretization for fixed ν and P_{avg} . Finally, Fig. 5a shows that HCSO finds solutions that reduce the objective function value by up to 42%, when compared to non-hierarchical CSO.]

Furthermore, as shown in Fig. 5b, bench-marking our MAESTRO-X framework against both model-based [13] and model-free [14] solutions in the state-of-the-art, we observe that the decoupled policy replication across the swarm—with suitable supplementary heuristics—ensures that our solution constitutes a practically feasible formulation even with higher-degrees of scaling

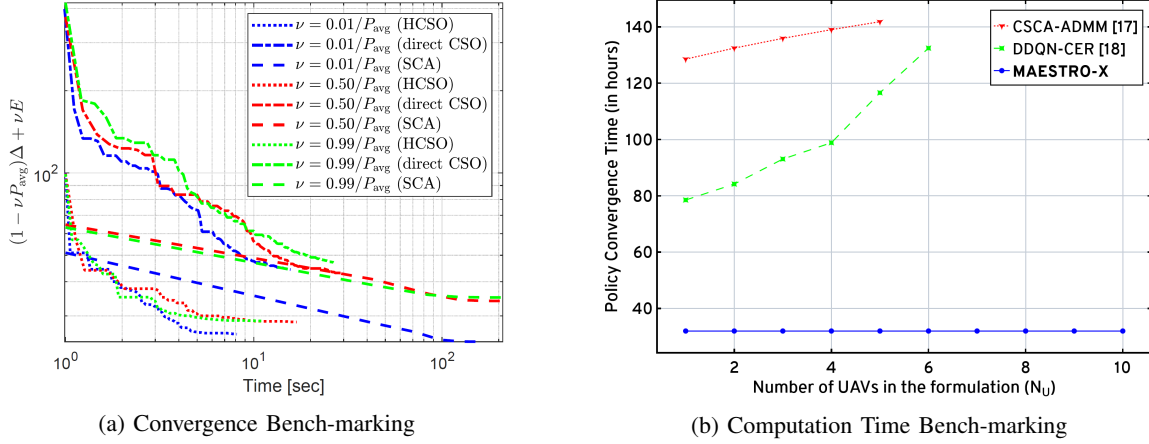


Fig. 5: Convergence bench-marking of HCSO against the CSO and SCA schemes from the state-of-the-art (a); Computational complexity bench-marking of MAESTRO-X against relevant frameworks in the state-of-the-art (b).

in the number of UAVs in the swarm. The Constrained Successive Convex Approximation (CSCA) wrapped Alternating Direction Method-of-Multipliers (ADMM) approach in [13] and the Double Deep-Q Network (DDQN) with Combined Experiential Replay (CER) approach in [14] involve prohibitively large policy convergence times when the number of the UAVs in the swarm is scaled beyond $N_U=5$ and $N_U=6$, respectively. All implementations are in Python, and this bench-marking is performed 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. The CSCA-ADMM solution in [13] suffers poor bench-marking performance relative to MAESTRO-X due to a joint multi-UAV construction involved in its CVXPY¹⁵-SCS¹⁶ implementation; the DDQN-CER¹⁷ solution in [14] demonstrates larger policy convergence times relative to MAESTRO-X due to an underlying model-free formulation, in addition to an MDP construction that considers combined multi-UAV state and action spaces. On the other hand, the decoupled policy executions in MAESTRO-X mitigates the computational overhead encountered by the joint/combined multi-UAV constructions in [13] and [14]; moreover, multi-threading concurrency (128 software threads per CPU) and distributed computing (over the 2 CPUs and 4 GPUs) facilitate accelerations in MAESTRO's policy convergence.

For our evaluations of the optimal *waiting* phase behavior, we fix $P_{\text{avg}} = 1.2$ kW and $L = 1$ Mb or 100 Mb. Figures 6a and 6b show that the UAV moves to and waits by flying around

¹⁵CVXPY: A library for solving convex optimization problems in Python.

¹⁶Split Conic Solver (SCS): The optimizer employed within our CVXPY implementation of [13].

¹⁷A TensorFlow-Keras implementation, the source code for which was made available on GitHub [14].

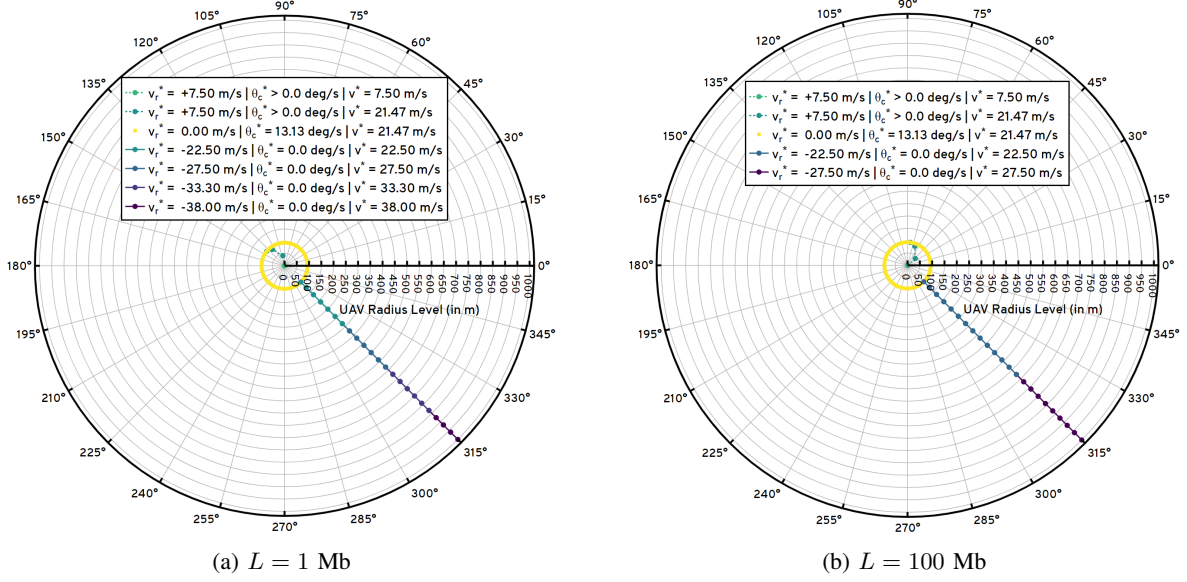
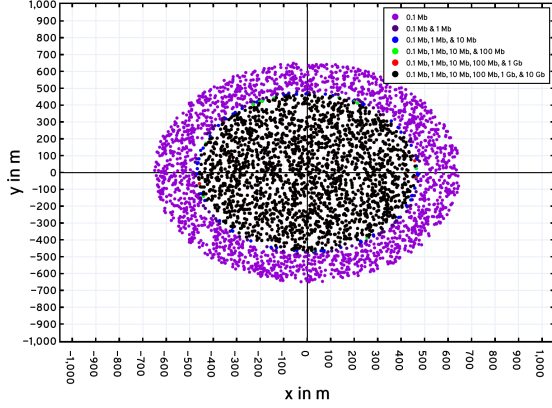


Fig. 6: Visualizations of MAESTRO's optimal waiting state policy for $L=1$ Mb and $L=100$ Mb, with $P_{\text{avg}} = 1.2$ kW fixed across both evaluations.

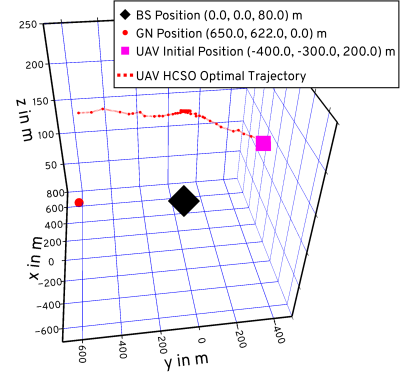
an optimal radius level (≈ 95 m from the center) to address two considerations: 1) to be well-positioned for future transmission requests; 2) to fly at the power-minimizing velocity so as to reduce UAV energy consumption. Comparing the UAV radial velocities between the $L = 1$ Mb and $L = 100$ Mb, we observe that, near the cell edge, the UAV radial velocity increases for decreasing data payload, as smaller data packets relaying through the UAV complete transmission faster than large ones—thus the UAV is less likely to spend time following trajectories that place it near the cell edge and can afford the additional power consumption required to fly at high velocities when this is the case.

Analyzing optimal decision-making in the *request scheduling* phase vis-à-vis the single-relay specialization of MAESTRO, Fig. 7a shows which GNs transmit directly to the BS under the optimal policy $\tilde{\mu}$ during these *request scheduling* phases. The target average power is $P_{\text{avg}} = 1.2$ kW, and the UAV radius is $r_U = 500$ m. One third of the GNs transmit directly to the BS for small data payloads, i.e., 0.1 Mb, and as the data payload grows larger and larger, the benefit of relaying through the UAV and utilizing its mobility increases, so that the majority of GNs relay through the UAV. Next, in Fig. 7b, we observe the trajectory of 3 UAVs in the swarm executed toward serving 3 different GN transmission requests. [\[BK: Re-generating Fig. 7b to visualize 2D trajectories of all 3 UAVs, each serving a different GN request.\]](#)

To analyze the communication delay performance, we simulate over various data payloads and target average power constraints, P_{avg} , to observe a range of behavior. The expected average delay



(a) Map of GNs with $\xi=0$ for $P_{\text{avg}} = 1.2\text{ kW}$



(b) Optimal trajectory for $L=10\text{ Mb}$ and $P_{\text{avg}} = 1.2\text{ kW}$

Fig. 7: The GNs that are scheduled to be handled by the BS according to MAESTRO's optimal policy in the *communication* states (a); The optimal trajectory of UAV₂ under the MAESTRO-X framework, as determined by HCSO, when the active GN request is at (650, 622, 0) m and UAV₂ is initially at (-400, -300, 200) m.

is shown in Figs. 8 and 9. Note that the expected average delay, $\bar{D}_{\tilde{\mu}}$, is not directly accessible through the value iteration analysis, so we apply the optimal policy $\tilde{\mu}$ in a simulation, sampling a sequence of 10,000 random transmission requests, in order to obtain it. In Fig. 8, we compare the performance of MAESTRO-X with that of different network deployment strategies, for $L = 1\text{ Mb}$ and $L = 100\text{ Mb}$ under varying average per-UAV power constraints: these evaluations are listed below.

- **M/G/x BS Only:** In this deployment, there are no UAV relays supplementing the terrestrial BS in the cell. We analyze the communication delays in this BS-only setting, in addition to queue wait times studies¹⁸ with $N_B = 1, 5,$ and 10 . Expectedly, we find a significant reduction in the average communication delay experienced by the GNs in the cell, by employing UAVs to relay data traffic. Additionally, as illustrated in the delay histograms, an increase in the number of orthogonal BS channels leads to a considerable increase in the queue wait times.
- **Static UAVs:** In this deployment, we evaluate the delay performance of static UAVs serving as a BS relay for the GNs in the cell. Here, we find that dynamic UAV deployments with optimized trajectories (way-points and velocities) demonstrate lower service delays compared to static deployments—specifically, for a swarm of 3 UAV relays, with $L = 100\text{ Mb}$ and a per-UAV power consumption of 1.37 kW , MAESTRO-X services uplink transmission

¹⁸All queueing studies in our numerical evaluations are performed via SimPy implementations in Python: M/G/ N_B for the BS and M/G/1 for each UAV.

requests from the GNs $20\times$ faster than a static deployment of 3 UAVs positioned equidistant from the cell center.

- **MAESTRO- α :** Here, we study a variant of MAESTRO-X, where in addition to the optimal *waiting* and *communication* policy extended from MAESTRO, with spread maximization and conflict resolution, we employ a ‘Join-Shortest-Queue (JSQ)¹⁹’ heuristic at the BS to determine the queue wait times during the consensus-driven conflict resolution process in the *request scheduling phase* as opposed to a ‘Join-Fastest-Queue (JFQ)²⁰’ heuristic in MAESTRO-X. For a swarm with 3 UAV relays, We observe that a JFQ-scheme at the BS results in $3\times$ lower average service delay than a JSQ-scheme.
- **MAESTRO- β :** Here, we study another variant of MAESTRO-X, where the optimal *waiting* state policy of MAESTRO is replaced with a heuristic in which the UAVs return to the cell center (at V_{\max}) immediately after transitioning to the *waiting* state; MAESTRO’s optimal *communication* policy is extended with consensus-driven conflict resolution as is the case for MAESTRO-X. We note that, for a swarm of 3 UAVs, with $L = 100$ Mb, this ‘Return-to-Center (R2C)’ *waiting* policy in MAESTRO- β as opposed to the optimal *waiting* policy in MAESTRO-X, results in $3\times$ worse delay performance.

Continuing with our delay-power analyses of orchestration frameworks for relay swarms, we outline our performance comparisons with relevant schemes in the current literature. These comparisons are depicted in Fig. 9. For a swarm of 3 UAV relays, with uplink transmission requests of size $L = 100$ Mb from the GNs in the cell, we observe the following improvements in performance over related schemes in the state-of-the-art, averaged over 10,000 requests with a traffic arrival rate of 1 request every 1800 s.

- **Successive Convex Approximation (SCA) [12]:** Here, with a CVXPY implementation involving 10^6 iterations and an accuracy of 10^{-6} , this SCA scheme relays GN requests $6\times$ slower than MAESTRO-X.
- **Constrained Successive Convex Approximation with Alternating Direction Method of Multipliers (CSCA-ADMM) [13]:** With an ADMM scheme employed within SCA, centered around linearly-coupled equality constraints, the framework proposed in [13] involves 50

¹⁹JSQ: The BS queue wait times for a new GN uplink transmission is determined based on the queue (channel) with the smallest number of requests in waiting

²⁰JFQ: The BS queue wait times for a new GN uplink transmission is determined based on the queue (channel) that has the smallest waiting time.

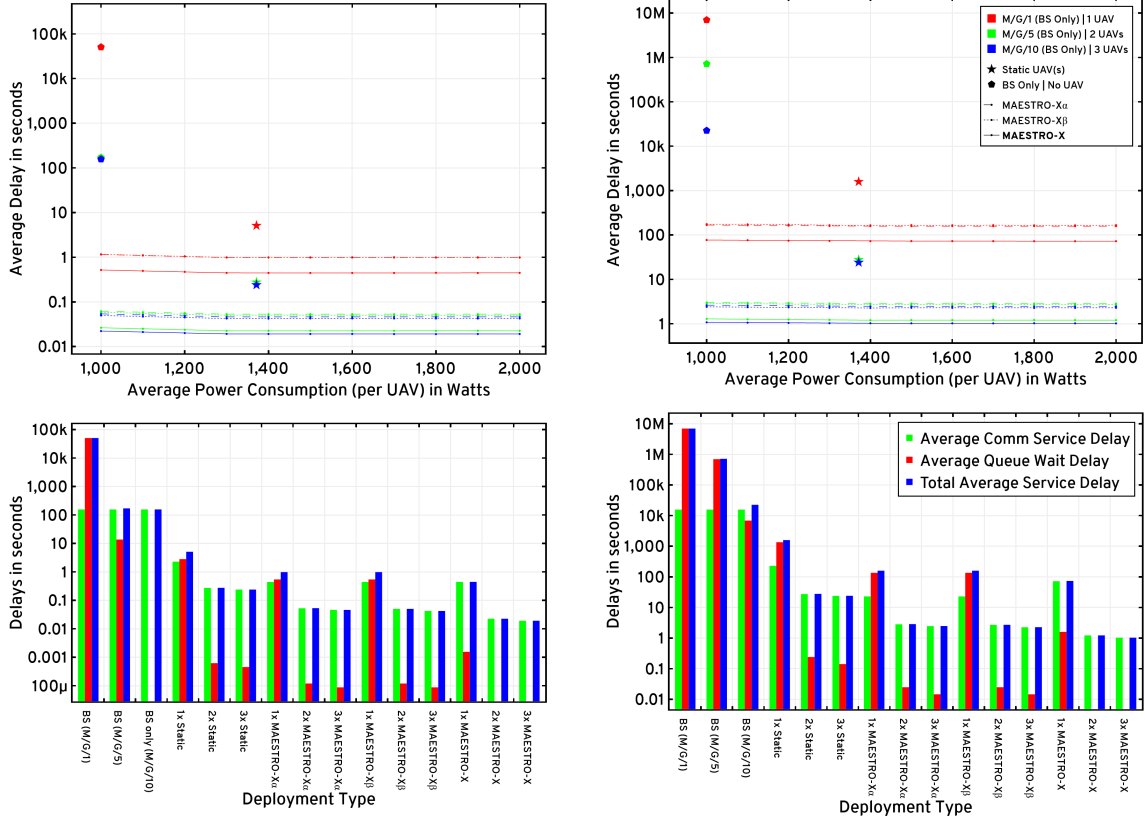


Fig. 8: The traces and histograms constituting the average service latencies (communication + queue wait) versus the average power constraint for the proposed MAESTRO-X framework under $L = 1$ Mb (left) and $L = 100$ Mb (right): Comparisons with custom policy heuristics.

worker threads (each with a Split Conic Solver optimizer, 10^6 iterations, and an accuracy of 10^{-6}) solving the optimization problem within an SCA wrapper; MAESTRO-X demonstrates $10\times$ better QoS performance vis-à-vis service latencies, for an average per-UAV power constraint of 1.2 kW.

- Double Deep Q-Network with Combined Experiential Replay (DDQN-CER) [14]: This model-free approach revolving around a decentralized POMDP²¹ formulation, with an average per-UAV power constraint of 1.1 kW demonstrates $4\times$ slower request service times relative to MAESTRO-X.

VII. MAESTRO-X ON AERPAW: FEASIBILITY ANALYSES

[BK: [Emulations underway](#): The emulations of MAESTRO and MAESTRO-X are in progress on the EVMs of the AERPAW testbed. Upon their completion, I will add a couple of images from the QGControl trajectory executions and talk about a few metrics from the traffic logs.]

²¹Partially Observable Markov Decision Processes

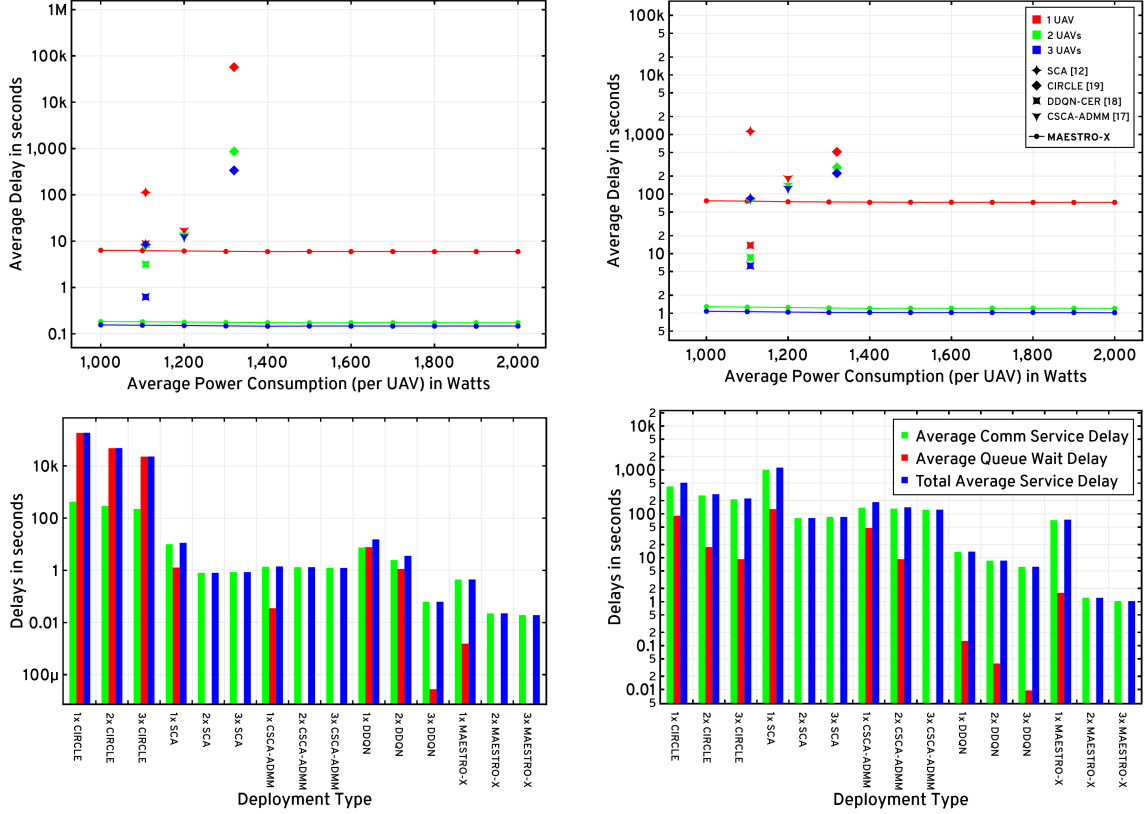


Fig. 9: The traces and histograms constituting the average service latencies (communication + queue wait) versus the average power constraint for the proposed MAESTRO-X framework under $L = 1$ Mb (left) and $L = 100$ Mb (right): Comparisons with relevant schemes in the state-of-the-art.

VIII. CONCLUSIONS

In this paper, we propose MAESTRO-X: a framework for the decentralized orchestration of a swarm of rotary-wing UAV relays in hybrid wireless networks, augmenting the coverage and service capabilities of a terrestrial BS. First, we specialize our system model to single-UAV deployments and design the optimal request scheduling and trajectory optimization policy under an SMDP formulation (via value iteration and HCSO). Next, we extend this single-relay policy to distributed deployments of two or more UAVs by supplementing the policy with multi-UAV coordination heuristics and M/G/x queueing analyses, and replicate this augmented policy across the swarm. Numerical evaluations demonstrate that MAESTRO-X delivers significant performance improvements to ‘BS-only’ and ‘Static UAV’ deployments; furthermore, MAESTRO-X outperforms popular Q-learning and successive convex approximation strategies in the existing literature. Finally, we demonstrate the integration and implementation viability of MAESTRO-X on the NSF AERPAW platform through QGroundControl based emulations.

APPENDIX A: PROOF OF LEMMA 1

Consider \bar{W}_μ in (8). 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) . Hence, the expected average communication delay to serve these additional requests is $\mathbb{E}[\Delta_{u,i}^{(bs)}] = \bar{D}_{BS}$, yielding

$$\bar{W}_\mu = \lim_{t \rightarrow \infty} \mathbb{E}_\mu \left[\frac{1}{M_t} \sum_{u=1}^{M_t} [\Delta_u^{(s)} + \xi_u N_u \bar{D}_{BS}] \right] = \bar{D}_{BS}(\bar{N}_\mu - 1) + \bar{W}_\mu^{(s)}, \quad (33)$$

where the second step uses the definition of \bar{N}_μ in (8). Then, it follows that

$$\bar{D}_\mu = \frac{\bar{W}_\mu}{\bar{N}_\mu} = \frac{1}{\bar{N}_\mu} \bar{W}_\mu^{(s)} + \left(1 - \frac{1}{\bar{N}_\mu}\right) \bar{D}_{BS}. \quad (34)$$

Let μ_{BS} be the policy such that the UAV flies at the power-minimizing velocity V_{\min} , and all requests are served by the BS. This policy is feasible (it uses minimum power) and its average delay is $\bar{D}_{\mu_{BS}} = \bar{D}_{BS}$, since all requests are served directly by the BS. Therefore, under the optimal policy we must have $\bar{D}_\mu^* \leq \bar{D}_{\mu_{BS}} = \bar{D}_{BS}$. Using (34) and the fact that $\bar{N}_\mu \geq 1$, this implies that $\bar{W}_\mu^{(s)} \leq \bar{D}_{BS}$. Let μ be any policy that satisfies $\bar{W}_\mu^{(s)} \leq \bar{D}_{BS}$ (note that any other policy is sub-optimal). Under such policy, (34) along with $\bar{N}_\mu \geq 1$, implies $\bar{W}_\mu^{(s)} \leq \bar{D}_\mu \leq \bar{D}_{BS}$.

Moreover, if $\xi_u=1$, then N_u requests are received during the UAV relay phase of duration $\Delta_u^{(s)}$; since these requests are received with a rate Λ , we find that $\mathbb{E}[N_u | \Delta_u^{(s)}] = \Delta_u^{(s)} \Lambda$, so that using the bound $\xi_u \leq 1$, we can express

$$\bar{N}_\mu = \lim_{t \rightarrow \infty} \mathbb{E}_\mu \left[\frac{1}{M_t} \sum_{u=1}^{M_t} (1 + \xi_u \Delta_u^{(s)} \Lambda) \right] \leq 1 + \Lambda \bar{W}_\mu^{(s)}, \quad (35)$$

with strict equality if the UAV always serves requests. Under any policy μ such that $\bar{W}_\mu^{(s)} \leq \bar{D}_{BS}$, we can then bound as in (10), where the second inequality follows from $\bar{W}_\mu^{(s)} \leq \bar{D}_{BS}$.

APPENDIX B: EVALUATION OF π_{comm}

Lemma 2. Let $\pi_{\text{comm}} = \int_{S_{\text{comm}}} \Pi_\mu(s) ds$ and $\pi_{\text{wait}} = 1 - \pi_{\text{comm}}$ be the steady-state probabilities that the UAV is in the waiting and communication phases in the SMDP, respectively. We have that

$$\pi_{\text{wait}} = \frac{1}{2 - e^{-\Lambda \Delta_0}}, \quad \pi_{\text{comm}} = \frac{1 - e^{-\Lambda \Delta_0}}{2 - e^{-\Lambda \Delta_0}}. \quad (36)$$

Proof. Let $p_{ww} = e^{-\Lambda \Delta_0}$ and $p_{cw} = 1$ be the probabilities of remaining in the waiting phase (ww ; if no request is received, the SMDP remains in the waiting state) and of moving from a communication to a waiting state (cw ; a waiting state always follows a communication state)

in one SMDP state transition, respectively. Therefore, π_{wait} and π_{comm} satisfy the stationary equations

$$\begin{cases} \pi_{\text{wait}} = \pi_{\text{wait}}p_{ww} + \pi_{\text{comm}}p_{cw} = e^{-\Lambda\Delta_0}\pi_{\text{wait}} + \pi_{\text{comm}}, \\ \pi_{\text{comm}} = \pi_{\text{wait}}(1 - p_{ww}) + \pi_{\text{comm}}(1 - p_{cw}) = (1 - e^{-\Lambda\Delta_0})\pi_{\text{wait}}, \text{ and} \\ \pi_{\text{wait}} + \pi_{\text{comm}} = 1; \end{cases}$$

whose solution is given in the statement of the lemma. ■

REFERENCES

- [1] M. Bliss and N. Michelusi, “Power-Constrained Trajectory optimization for Wireless UAV Relays with Random Requests,” in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–6.
- [2] B. Keshavamurthy, “Aesir: An Adaptive Intelligent Distributed and Scalable UAV Fleet Orchestration Framework,” March 2022. [Online]. Available: <https://github.com/bharathkeshavamurthy/Aesir.git>
- [3] A. Fotouhi, H. Qiang, M. Ding, M. Hassan, L. G. Giordano, A. Garcia-Rodriguez, and J. Yuan, “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, M. Bennis, Y. Nam, and M. Debbah, “A Tutorial on UAVs for Wireless Networks: Applications, Challenges, and Open Problems,” *IEEE Communications Surveys Tutorials*, pp. 1–1, 2019.
- [5] Y. Zeng, R. Zhang, and T. J. Lim, “Wireless communications with unmanned aerial vehicles: opportunities and challenges,” *IEEE Communications Magazine*, vol. 54, no. 5, pp. 36–42, May 2016.
- [6] Q. Wu, L. Liu, and R. Zhang, “Fundamental Trade-offs in Communication and Trajectory Design for UAV-Enabled Wireless Network,” *IEEE Wireless Communications*, vol. 26, pp. 36–44, 02 2019.
- [7] B. Keshavamurthy and N. Michelusi, “Learning-based Spectrum Sensing and Access in Cognitive Radios via Approximate POMDPs,” *IEEE Transactions on Cognitive Communications and Networking*, 2021.
- [8] DARPA-SC2-Freshdesk-CE, “Alleys of Austin with Points - 5 Teams,” *DARPA Spectrum Collaboration Challenge (SC2)*, 2019. [Online]. Available: <https://sc2colosseum.freshdesk.com/support/solutions/articles/22000237879-pe2-alleys-of-austin-w-points-5-team-7013->
- [9] V. C. Inc., “Autonomous Drone Surveillance,” 2021. [Online]. Available: <https://www.verizon.com/business/resources/5g/5g-business-use-cases/public-sector/autonomous-drone-surveillance/>
- [10] D. Alvear, “Inside Verizon | Up to Speed | Drones, robots and the power of 5G,” July 2021. [Online]. Available: <https://www.verizon.com/business/resources/5g/5g-business-use-cases/public-sector/autonomous-drone-surveillance/>
- [11] S. Guillette, “Verizon News Archives | Robots, drones and sensors are changing the way we farm,” March 2019. [Online]. Available: <https://www.verizon.com/business/resources/5g/5g-business-use-cases/public-sector/autonomous-drone-surveillance/>
- [12] Y. Zeng, J. Xu, and R. Zhang, “Energy Minimization for Wireless Communication With Rotary-Wing UAV,” *IEEE Transactions on Wireless Communications*, vol. 18, no. 4, pp. 2329–2345, April 2019.
- [13] Q. Hu, Y. Cai, A. Liu, G. Yu, and G. Y. Li, “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, pp. 6179–6195, 2020.

- [14] H. Bayerlein, M. Theile, M. Caccamo, and D. Gesbert, "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. [Online]. Available: <http://dx.doi.org/10.1109/OJCOMS.2021.3081996>
- [15] 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.
- [16] X. Hu, K.-K. Wong, K. Yang, and Z. Zheng, "UAV-Assisted Relaying and Edge Computing: Scheduling and Trajectory Optimization," *IEEE Transactions on Wireless Communications*, vol. 18, no. 10, pp. 4738–4752, 2019.
- [17] L. Wang, K. Wang, C. Pan, W. Xu, N. Aslam, and L. Hanzo, "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, pp. 73–84, 2021.
- [18] 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.
- [19] A. E. A. A. Abdulla, Z. M. Fadlullah, H. Nishiyama, N. Kato, F. Ono, and R. Miura, "Toward Fair Maximization of Energy Efficiency in Multiple UAS-Aided Networks: A Game-Theoretic Methodology," *IEEE Transactions on Wireless Communications*, vol. 14, no. 1, pp. 305–316, Jan 2015.
- [20] 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.
- [21] Q. Wu, Y. Zeng, and R. Zhang, "Joint Trajectory and Communication Design for Multi-UAV Enabled Wireless Networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 3, pp. 2109–2121, 2018.
- [22] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Efficient Deployment of Multiple Unmanned Aerial Vehicles for Optimal Wireless Coverage," *IEEE Communications Letters*, vol. 20, no. 8, pp. 1647–1650, 2016.
- [23] J. Hu, H. Zhang, and L. Song, "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.
- [24] A. M. Koushik, F. Hu, and S. Kumar, "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, pp. 554–566, 2019.
- [25] Q. Zhang, M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Machine Learning for Predictive On-Demand Deployment of UAVs for Wireless Communications," in *2018 IEEE Global Communications Conference (GLOBECOM)*, 2018, pp. 1–6.
- [26] 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.
- [27] 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.
- [28] M. Clerc, *Particle Swarm Optimization*, ser. ISTE. London ; Newport Beach: ISTE, 2006.
- [29] H. Shakhathreh, A. Khreishah, A. Alsarhan, I. Khalil, A. Sawalmeh, and N. S. Othman, "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.
- [30] Z. Yuheng, Z. Liyan, and L. Chunpeng, "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.
- [31] R. K. Patra and P. Muthuchidambaranathan, "Optimisation of Spectrum and Energy Efficiency in UAV-Enabled Mobile Relaying Using Bisection and PSO Method," in *2018 3rd International Conference for Convergence in Technology (I2CT)*, 2018, pp. 1–7.

- [32] V. Marojevic, I. Guvenc, R. Dutta, M. L. Sichitiu, and B. A. Floyd, "Advanced Wireless for Unmanned Aerial Systems: 5G Standardization, Research Challenges, and AERPAW Architecture," *IEEE Vehicular Technology Magazine*, vol. 15, no. 2, pp. 22–30, 2020.
- [33] B. Keshavamurthy, Y. Zhang, C. R. Anderson, N. Michelusi, J. V. Krogmeier, and D. J. Love, "A Robotic Antenna Alignment and Tracking System for Millimeter Wave Propagation Modeling," 2021. [Online]. Available: <https://arxiv.org/abs/2110.07106>
- [34] A. Visintini, T. D. P. Perera, and D. N. K. Jayakody, "3-D Trajectory Optimization for Fixed-Wing UAV-Enabled Wireless Network," *IEEE Access*, vol. 9, pp. 35 045–35 056, 2021.
- [35] S. Hanna, E. Krijestorac, H. Yan, and D. Cabric, "UAV Swarms as Amplify-and-Forward MIMO Relays," in *2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2019, pp. 1–5.
- [36] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal LAP Altitude for Maximum Coverage," *IEEE Wireless Communications Letters*, vol. 3, no. 6, pp. 569–572, 2014.
- [37] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge: Cambridge University Press, 2005.
- [38] 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.
- [39] M.-H. Hwang, H.-R. Cha, and S. Jung, "Practical endurance estimation for minimizing energy consumption of multirotor unmanned aerial vehicles," *Energies*, vol. 11, no. 9, 2018.
- [40] J. D. C. Little and S. Graves, *Little's Law*, 07 2008.
- [41] S. Boyd, L. Xiao, and A. Mutapcic, "Subgradient methods," *Lecture notes, Stanford University, Autumn Quarter*, vol. 2004, pp. 2004–2005, 2003.
- [42] 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.
- [43] P. Yang, K. Tang, and X. Yao, "Turning High-Dimensional Optimization Into Computationally Expensive Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 143–156, 2018.
- [44] TensorFlow, "The TensorFlow Blog," *TensorFlow Blog*, 2021. [Online]. Available: <https://blog.tensorflow.org/>
- [45] TensorFlow Core, "Use a GPU | TensorFlow Core," *TensorFlow Documentation*, 2022. [Online]. Available: <https://www.tensorflow.org/guide/gpu>
- [46] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal LAP Altitude for Maximum Coverage," *IEEE Wireless Communications Letters*, vol. 3, no. 6, pp. 569–572, 2014.