Multi-UAV Energy-Efficient Wildfire Coverage Optimization

Carles Diaz-Vilor, Mohammadreza Barzegaran, and Hamid Jafarkhani

***Abstract*—Uncrewed aerial vehicles (UAVs) are expected to play a pivotal role in 6G networks due to their versatility and adaptability. One potential application for UAVs is wildfire coverage, as they can carry various sensors, including cameras and antennas. This study focuses on the multi-UAV trajectory optimization for wildfire coverage while satisfying multiple constraints, including the UAV dynamics, network connectivity, and limited energy batteries. The resulting complex optimization problem is time-varying and non-convex. To address this challenge, reinforcement learning, specifically the twin-delayed deep deterministic policy gradient algorithm, is adopted. A distributed learning procedure is devised to allow parallelization and significant reduction of the training time. The result is high coverage at standard flying altitudes with finite energy batteries.**

***Index Terms*—UAV, wildfire coverage, energy efficiency, drone dynamics, age of information, reinforcement learning, TD3**

I. Introduction

Uncrewed aerial vehicles (UAVs) offer a versatile solution for enhancing communications, sensing, and data collection in 6G networks. Their inherent mobility and adaptability enable dynamic deployment and flexibility across various environments and contexts. The enhanced coverage enabled by UAVs, coupled with the ultra-high data rates anticipated from 6G networks, presents an application poised to revolutionize a multitude of sectors. Precisely, equipped with advanced sensors, cameras and communication equipment, UAVs efficiently collect real-time data and provide connectivity to remote or inaccessible regions where wildfires pose an immediate threat. This aerial perspective allows for early detection of fire outbreaks, accurate mapping of fire perimeters, monitoring of fire behavior and progression in real time and better understanding their spread, which can help identifying areas at risk [1]–[3]. By integrating UAVs into 6G networks, we can further enhance wildfire tracking capabilities, enabling seamless communication and coordination among firefighting teams while facilitating timely and effective response strategies to mitigate the impact of wildfires [4]–[7].

Recently, there has been significant interest in studying UAV deployment and trajectory for various applications, with wildfire monitoring and sensing emerging as crucial areas [8]–[13]. However, the design of such trajectories is contingent to three main ingredients: (*i*) the UAV dynamics, (*ii*) the UAVs' constrained energy sources, and (*iii*) the wireless transmission of the captured video footage to the network.

The most crucial aspect of UAV trajectory design is considering their dynamics, which is often avoided in the literature. UAVs, like other robots, possess limited maneuvering capabilities. These limitations may stem from various factors including the dynamical behavior, actuation constraints, and control robustness and performance [14]. In fact, without carefully designing their trajectories, UAVs may attempt maneuvers that exceed their capabilities, resulting in instabilities, unpredictable behavior, collisions, or crashes [15], [16]. To this end, researchers have proposed a number of approaches ensuring that the planned trajectories remain within the maneuverability limits of the UAVs [17]–[21]. To be precise, this work utilizes UAV tracking dynamics to predict their behavior. Our proposed approach only determines trajectories that are rendered safe by UAV tracking dynamics.

Another key feature when designing UAV trajectories is the limited available energy. In addition to safe operation, the mission's success depends on determining trajectories that account for energy consumption. For that, various energy consumption models exist in the literature [22]–[24]. In fact, this paper extends the relationship between the UAVs movements and their energy consumption by including their dynamic behavior. To this end, we identify various parameters influencing the UAV energy requirements based on their dynamics. Subsequently, an energy consumption model is developed to capture different trajectory aspects. Our proposed approach utilizes this energy consumption model to ensure energy-efficient trajectories. In addition, we explore scenarios where charging stations are available. To be precise, our proposed approach estimates the energy needed to fly to the charging stations. When the remaining battery energy is only adequate to reach a charging station, UAVs are directed to the charging stations.

The last related key feature in designing the UAV trajectories is the wireless transmission of the captured video or images. Correct reception of such information is a necessary aspect of wildfire coverage. In fact, optimizing UAV trajectories in wireless networks has become a focal point of research, as UAVs can serve various roles such as flying base stations (BSs), relays, users, or sensors [25]–[46], where all these references avoid the UAV dynamics. Despite this, the combination of multi-UAV trajectory optimization for wildfire coverage under cellular wireless connectivity, as done in this work, remains largely unexplored. After capturing an image, UAVs concurrently transmit them to the BS. UAV transmissions may experience delays if the UAVs are far from the BS or encounter significant interference. We consider the age of information (AoI) as the metric to measure data freshness [47], where a small AoI indicates low delay and a large AoI means that the captured images reach the BS with a high delay. The goal is to adjust UAV trajectories and transmit powers to keep the AoI at bay. In fact, we have considered the problem of UAV trajectory optimization for a cell-free wireless

The authors are with Center for Pervasive Communications and Computing, University of California, Irvine CA 92697, USA. This work was supported in part by NSF Award CNS-2209695. Emails: {cdiazvil, barzegm1, hamidj}@uci.edu

network [48]. However, there are fundamental differences between this manuscript and the work in [48]: (*i*) the current work features cellular connectivity and considers AoI whereas our previous paper considers cell-free networks, (*ii*) the UAV dynamics are ignored in [48] while this paper includes them, (*iii*) the swarm's cost function is different, and (*iv*) this article derives and includes a new UAV energy consumption model.

The resulting optimization problem is challenging to solve because of (*i*) the wildfire's time-varying nature, (*ii*) the complexity of UAV dynamics, (*iii*) UAV-to-UAV interference, and (*iv*) the non-convexity with respect to most of the involved functions. In fact, traditional methods such as projected gradient ascent or the successive convex approximation (SCA) technique would provide far-from-optimal solutions. Therefore, this work leverages reinforcement learning (RL), where the goal is to learn optimal policies upon interacting with an environment [49]. Additionally, distributed learning can highly reduce the training time and complexity without compromising optimality [50], [51]. This manuscript decomposes the main problem into simpler components and utilizes deep Q-learning (DQL) to solve each one individually [52]–[56]. In fact, a variety of DQL algorithms have been recently used for UAV trajectory optimization purposes [38]–[42], where one solution stands out among all: the so-called twin-delayed deep deterministic policy gradient (TD3) [57] which has shown to achieve outstanding performances within the UAV framework [43]–[46]. TD3 is an effective solution for non-convex optimization problems with continuous variable domains, as it can capture complex, non-linear cost functions. Unlike gradient-based methods, TD3 uses reinforcement learning to optimize without requiring explicit gradients, allowing it to navigate challenging landscapes with multiple local minima. It also directly optimizes real cost functions without the need for approximations, making it well-suited for complex, real-world problems where model knowledge or simulations are limited. Therefore, we leverage the TD3 algorithm to solve the wildfire coverage problem under communications, dynamics, and energy constraints. Table I compares the applications of different UAV trajectory optimization works and their proposed solutions. To the best of our knowledge, this is the first work considering a number of energy-efficient UAVs for tracking wildfire under cellular MIMO AoI constraints.

The paper's contributions are:

- An analytical framework for wildfire coverage with UAVs equipped with cameras is introduced. In addition, the UAV dynamics are accounted for and a novel energy consumption is derived. Moreover, the cellular communication between UAVs and the network must satisfy a maximum latency requirement.
- By combining dynamical, communications, and finite-energy constraints, the multi-UAV energy efficient wildfire coverage problem is defined, where the optimization is performed with respect to the UAV trajectories and transmit powers.
- The TD3 algorithm is leveraged to solve the problem with respect to the UAV trajectories and transmit powers. The dependency of the wildfire coverage with respect to the number of UAVs, their available energy, and the flying altitude is studied.

The remainder of the manuscript is organized as follows. Sec. II presents the UAV dynamics, wildfire spread, camera, and communication models. Next, in Sec. III, the wildfire coverage problem is formulated. Sec. IV focuses on the RL-based solution while numerical results are discussed in Sec. V. Concluding remarks are provided in Sec. VI.

Notation: Small letters, bold letters, and bold capital letters designate scalars, vectors, and matrices, respectively. Matrices $\boldsymbol{A}^{\mathrm{T}}$ and $\boldsymbol{A}^{\mathrm{H}}$ are the transpose and the Hermite transpose of matrix $\boldsymbol{A}$, respectively.

II. System Models

Consider a system with M UAVs equipped with video surveillance cameras. The m^{th} UAV's 3D position and velocity are denoted by $\boldsymbol{q}_m^{(n)} = \big(x_m^{(n)}, y_m^{(n)}, h_m^{(n)}\big)$ and $\dot{\boldsymbol{q}}_m^{(n)} = \big(\dot{x}_m^{(n)}, \dot{y}_m^{(n)}, \dot{h}_m^{(n)}\big)$, respectively, where n is the time index. The duration of each time slot is δ. UAVs connect to the core network via a cellular BS located at coordinates $\boldsymbol{q}_{\mathrm{b}} = (x_{\mathrm{b}}, y_{\mathrm{b}}, h_{\mathrm{b}})$. Additionally, there are L charging stations distributed across the region of interest, with the ℓ^{th} station located at $\boldsymbol{q}_\ell^{\mathrm{c}} = (x_\ell^{\mathrm{c}}, y_\ell^{\mathrm{c}}, h_\ell^{\mathrm{c}})$. Finally, we consider a time-varying event of interest that the UAV cameras aim to cover. We denote the density of the events at a generic 2D point (x, y) at time n by $\psi(x, y, n)$. While our formulation and solution are applicable for any $\psi(x, y, n)$, this study assumes that this function represents the spread of a wildfire (see Sec. II-B).

A. UAV Dynamics

This study assumes all UAVs are commercial rotary-wing drones, each equipped with ξ propellers and a position-tracking controller. The UAV tracking dynamics are presented using a discrete-time state-space representation:

$$\big(\boldsymbol{q}_m^{(n+1)}, \dot{\boldsymbol{q}}_m^{(n+1)}\big)^{\mathrm{T}} = \boldsymbol{A}_m \big(\boldsymbol{q}_m^{(n)}, \dot{\boldsymbol{q}}_m^{(n)}\big)^{\mathrm{T}} + \boldsymbol{B}_m \boldsymbol{u}_m^{(n)}, \quad (1)$$

where $\boldsymbol{u}_m^{(n)} \in \mathbb{R}^3$ is the target position at a given time step n, also known as reference signal, and $\boldsymbol{A}_m \in \mathbb{R}^{6\times 6}$ captures how the current states, i.e., position and velocity, influence the next states. Similarly, $\boldsymbol{B}_m \in \mathbb{R}^{6\times 3}$ describes how $\boldsymbol{u}_m^{(n)}$ affects the next states. These matrices are unique to each UAV and are determined using system identification methods. Note that, in fact, the inclusion of such dynamics highly increases the difficulty of the problem given the non-linear relationships. For example, none of the references cited in Table I takes into account the dynamics of UAVs. In addition to the relationship presented in (1), each UAV has performance limitations on the maximum velocity and acceleration denoted by $\dot{q}_m^{\max}$ and $\ddot{q}_m^{\max}$, respectively. Thus, the following constraints must be met:

$$\|\dot{\boldsymbol{q}}_m\| \leq \dot{q}_m^{\max}, \quad \|\dot{\boldsymbol{q}}_m^{(n+1)} - \dot{\boldsymbol{q}}_m^{(n)}\| \leq \ddot{q}_m^{\max}\delta. \quad (2)$$

Nevertheless, our tracking dynamics model is general and can implement any control affine form represented by $\big(\boldsymbol{q}^{(n+1)}, \dot{\boldsymbol{q}}^{(n+1)}\big)^{\mathrm{T}} = f\big(\boldsymbol{q}^{(n)}, \dot{\boldsymbol{q}}^{(n)}\big) + g\big(\boldsymbol{u}^{(n)}\big)$.

B. Fire Propagation Model

The density function $\psi(x, y, n)$ is tailored to the observed event, with a primarily focus on wildfire coverage in this

TABLE I: Comparing the characteristics of different UAV trajectory optimization methods. SCA: successive convex approximation; SAC: soft actor-critic; GT: graph theory; DP: dynamic programming; GA: gradient ascent; A3C: asynchronous actor critic; KF: Kalman filter.

	Multi-UAV	Energy-efficiency	Application	MIMO	AoI	Solution
[27]	✗	✓	Sensing and Comms	✗	✗	SCA
[32]	✗	✗	Comms	✗	✗	SAC
[38]	✓	✗	Federated Learning	✗	✗	A3C
[39], [40]	✗	✓	Comms	✗	✗	DDPG, DRL
[41]	✓	✓	Target Tracking	✗	✗	SAC
[42]	✓	✗	Comms	✗	✗	DDPG
[43]	✓	✓	Data Collection	✗	✓	TD3
[45]	✗	✗	Data Collection	✗	✗	TD3
[58]	✗	✓	Comms	✗	✗	SAC
[59]	✗	✗	Data Collection	✗	✗	TD3
[60]	✗	✓	Data Collection	✗	✗	TD3, SCA, DP
This work	✓	✓	Wildfire Tracking	✓	✓	TD3

work. To simulate the fire perimeter, $\psi(x, y, n)$ is generated using the widely recognized FARSITE model [2], [3]. In FARSITE, the spread of each ignition follows an elliptical pattern, while subsequent growth points are determined by Huygens' principle [3]. These growth points collectively form a new front by computing the convex hull of the new ellipses and are influenced by various environmental factors such as weather conditions, including wind direction and speed, fuel types, and terrain characteristics. The major and minor axes of these ellipses, denoted by $2a^{(n)}$ and $2b^{(n)}$, respectively, are computed as

$$a^{(n)} = \frac{1}{2\,\mathsf{LB}^{(\mathsf{n})}}\left(\mathsf{R} + \frac{\mathsf{R}}{\mathsf{HB}^{(\mathsf{n})}}\right), \quad b^{(n)} = \frac{1}{2}\left(\mathsf{R} + \frac{\mathsf{R}}{\mathsf{HB}^{(\mathsf{n})}}\right), \tag{3}$$

where the fire's steady-state spread is R [m/min] and

$$\mathsf{LB}^{(\mathsf{n})} = 0.936\, e^{0.2566 U^{(n)}} + 0.461\, e^{-0.1548 U^{(n)}} - 0.397 \tag{4}$$

$$\mathsf{HB}^{(\mathsf{n})} = \frac{\mathsf{LB}^{(\mathsf{n})} + \sqrt{[\mathsf{LB}^{(\mathsf{n})}]^2 - 1}}{\mathsf{LB}^{(\mathsf{n})} - \sqrt{[\mathsf{LB}^{(\mathsf{n})}]^2 - 1}}. \tag{5}$$

In our context, $U^{(n)}$ [m/s] and $\theta^{(n)}_{\text{wind}}$ are defined as the midflame wind speed and direction, respectively, modeled as $U^{(n)} \sim |\mathcal{N}(U_0, \sigma_U)|$ and $\theta^{(n)}_{\text{wind}} \sim \mathcal{N}(\overline{\theta}_{\text{wind}}, \sigma_{\text{wind}})$. Therefore, the ellipse generated by the i^{th} front point at time $n+1$, denoted by $\boldsymbol{e}^{(n)}_i \in \mathbb{R}^2$, is given by:

$$\boldsymbol{e}^{(n)}_i + \delta \begin{bmatrix} c^{(n)}_x \sin\theta^{(n)}_{\text{wind}} + a^{(n)} \cos\omega \\ c^{(n)}_y \cos\theta^{(n)}_{\text{wind}} + b^{(n)} \sin\omega \end{bmatrix}, \tag{6}$$

where $c^{(n)}_x$ and $c^{(n)}_y$ denote the fire spreading gradients and $0 \le \omega \le 2\pi$. To maintain a general formulation without requiring extensive environmental details, such as weather conditions, fuels, and terrain, we adopt a simplified set of FARSITE parameters widely utilized in the literature [8], [9]. This simplified approach is based on [3] and assumes:

$$c^{(n)}_x = c^{(n)}_y = \frac{\mathsf{R}}{2}\left(1 - \frac{1}{\mathsf{HB}^{(n)}}\right). \tag{7}$$

To create $\psi(x, y, n)$, a 2D histogram is generated, assigning a nonzero weight to the points along the perimeter of the fire. To provide a visual representation of the propagation model, we include an example in Fig. 1. In this example, the wildfire starts from an ignition point (black dot). Assuming the wind blows in the direction of the blue arrow, the next fire front is represented by an ellipse with minor and major axes, $2a^{(n)}$ and $2b^{(n)}$, respectively, as illustrated in Fig. 1b. Each blue point on the new fire front acts as a new ignition point, and the convex hull formed by the corresponding ellipses generates the updated fire front shown in Fig. 1d. In this context, the density function $\psi(x, y, n)$ is associated with the perimeter, indicated by the red curve.

C. Camera Model

A camera's field of view (FoV) is the observable spatial extent of a planar space $\mathcal{F} \in \mathbb{R}^2$. Assuming a downward-facing camera on the m^{th} UAV with a yaw angle of $\beta^{(n)}_m$, the FoV is represented as a rectangular region $\mathcal{B}^{(n)}_m$, defined as

$$\mathcal{B}^{(n)}_m = \left\{(x, y) : \left|\boldsymbol{S}(\beta^{(n)}_m)\left(x^{(n)}_m - x, y^{(n)}_m - y\right)^{\mathrm{T}}\right| \le h^{(n)}_m \tan\boldsymbol{\alpha}\right\}, \tag{8}$$

where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2)^{\mathrm{T}}$ represents the two half-view angles (see Fig. 2) and $\boldsymbol{S}(\beta^{(n)}_m)$ is the rotation matrix associated to $\beta^{(n)}_m$, defined as

$$\boldsymbol{S}(\beta^{(n)}_m) = \begin{pmatrix} \cos\beta^{(n)}_m & \sin\beta^{(n)}_m \\ -\sin\beta^{(n)}_m & \cos\beta^{(n)}_m \end{pmatrix}. \tag{9}$$

Note that the addition of $\boldsymbol{S}(\beta^{(n)}_m)$ poses a challenge, as the calculation of the coverage requires a more complex search over the planar space. In fact, some of the previous works featuring UAVs with cameras assume non-rotated FoV [48], [61]. Hence, in the general case of employing M UAVs, the coverage provided by their cameras can be calculated as the ratio between the non-zero density points covered by UAVs and the total number of non-zero density points:

$$C^{(n)} = \frac{\int_{\mathcal{F}} \mathbb{I}\{\psi(x, y, n) > 0\}\, \mathbb{I}\left\{\exists \mathcal{B}^{(n)}_m : (x, y) \in \mathcal{B}^{(n)}_m\right\} dxdy}{\int_{\mathcal{F}} \mathbb{I}\{\psi(x, y, n) > 0\}\, dxdy}, \tag{10}$$

where (x, y) represents a generic point in $\mathcal{F}$ and $\mathbb{I}\{\cdot\}$ is an indicator function equal to one when the condition is met and

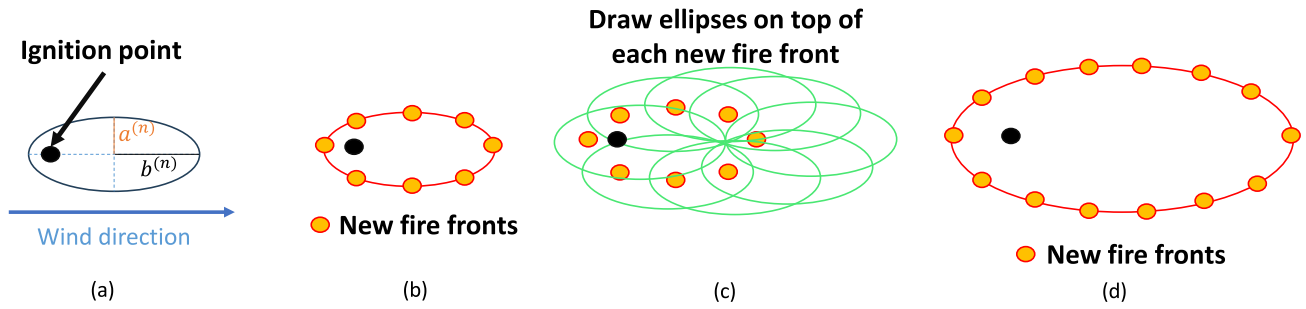


Fig. 1: Representation of the FARSITE fire propagation model.

zero otherwise. Note that there is a trade-off between coverage and image resolution; higher flying altitudes result in increased coverage at the expense of a lower resolution image and vice-versa. Consequently, the resolution of an image captured by the m^{th} UAV camera with I pixels can be quantified by computing the number of pixels per square meter:

$$\iota_m^{(n)} = \frac{I}{4\left(h_m^{(n)}\right)^2 \tan(\alpha_1)\tan(\alpha_2)}. \tag{11}$$

Therefore, by imposing a minimum image resolution denoted by $\iota_{\min}$, i.e., $\iota_m^{(n)} \geq \iota_{\min}$, we obtain a constraint on the maximum flying altitude as

$$h_m^{(n)} \leq \frac{1}{2\iota_{\min}} \sqrt{\frac{I}{\tan(\alpha_1)\tan(\alpha_2)}}. \tag{12}$$

The number of bits needed to transmit an image is denoted by B. For a 24-bit RGB system, after compressing the image by a factor of $\rho \in (0, 1]$, the number of required bits is

$$B = 24I\rho. \tag{13}$$

In addition to Eq. (12), local regulations and physical obstacles require UAVs to operate within designated altitude ranges. These minimum and maximum altitudes, denoted by $h_{\min}$ and $h_{\max}$, respectively, ensure safety, prevent collisions, and comply with airspace rules resulting in

$$h_{\min} \leq h_m^{(n)} \leq \min\left\{\frac{1}{2\iota_{\min}}\sqrt{\frac{I}{\tan(\alpha_1)\tan(\alpha_2)}}, h_{\max}\right\}. \tag{14}$$

D. Energy Consumption Model

In this paper, the UAV's energy consumption comprises three components[1]: (*i*) flying power $p_{m,\text{f}}^{(n)}$, (*ii*) processing power $p_{m,\text{p}}^{(n)}$, and (*iii*) communication power $p_m^{(n)}$. Thus, the energy consumption of the m^{th} UAV over a single time slot is

$$E_m^{(n)} = \delta\left(p_{m,\text{f}}^{(n)} + p_{m,\text{p}}^{(n)} + p_m^{(n)}\right). \tag{15}$$

The required flying power $p_{m,\text{f}}^{(n)}$ is in the form of electrical power used to rotate the propellers. Assuming ideal electric motors and neglecting orientation maneuvers, $p_{m,\text{f}}^{(n)}$ is composed by hovering power $p_{m,\text{h}}^{(n)}$ and kinetic power $p_{m,\text{k}}^{(n)}$ as

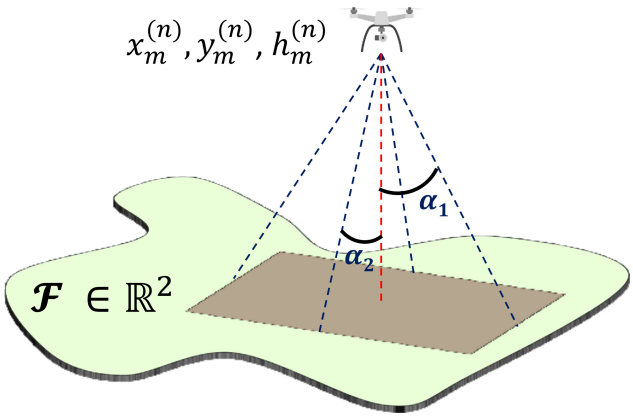


Fig. 2: Rectangular FoV of the m^{th} UAV for given α_1 and α_2 over a planar region $\mathcal{F}$.

$$p_{m,\text{f}}^{(n)} = p_{m,\text{h}}^{(n)} + p_{m,\text{k}}^{(n)}. \tag{16}$$

Building upon existing methods [62]–[65], we model hovering and kinetic power consumptions for our specific application. Using Appendix A, the hovering power consumption is

$$p_{m,\text{h}}^{(n)} = \frac{\mathsf{mg}\mathsf{C_D}}{\mathsf{C_L}} \sqrt{\frac{2\mathsf{mg}}{\xi\zeta\pi\mathsf{r}^2\mathsf{C_L}}}, \tag{17}$$

where, m is the UAV's mass, g is the gravitational acceleration, ζ is the air density, r is the propellers' length, $\mathsf{C_L}$ is the lift coefficient of the propellers, and $\mathsf{C_D}$ is the drag coefficient of the propellers. Similarly, using Appendix B, the kinetic power consumption is calculated as

$$p_{m,\text{k}}^{(n)} = \frac{2\pi\mathsf{m}\mathsf{C_M}}{\delta\mathsf{J}\mathsf{C_L}} \big(\underbrace{\dot{\boldsymbol{q}}_m^{(n+1)} - \dot{\boldsymbol{q}}_m^{(n)}}_{\Delta\dot{\boldsymbol{q}}_m^{(n)}}\big)^{\mathrm{T}} \dot{\boldsymbol{q}}_m, \tag{18}$$

where J and $\mathsf{C_M}$ are the propeller's geometric pitch and pitching coefficient, respectively. Ignoring thrust adjustment maneuvers, i.e., $\angle\big(\Delta\dot{\boldsymbol{q}}_m, \dot{\boldsymbol{q}}_m\big) = 0$, the kinetic power can be expressed as

$$p_{m,\text{k}}^{(n)} = \frac{2\pi\mathsf{m}\mathsf{C_M}}{\delta\mathsf{J}\mathsf{C_L}} \|\Delta\dot{\boldsymbol{q}}_m^{(n)}\| \|\dot{\boldsymbol{q}}_m\|. \tag{19}$$

Given that $p_{m,\text{p}}^{(n)}$ is typically much smaller than $p_{m,\text{f}}^{(n)}$, it can be assumed to be fixed, i.e., $p_{m,\text{p}}^{(n)} = p_m^p$, where p_m^p is

[1]It is acknowledged that there may be more components involved in the UAV's energy consumption related to their diverse functionalities and capabilities.

a constant. Finally, the amount of power each UAV requires for communicating with the BS is denoted by $p_m^{(n)}$ and satisfies

$$p_m^{(n)} \leq p_{\max}, \tag{20}$$

where $p_{\max}$ is the maximum transmit power.

Since $E_m^{(n)}$ denotes the m^{th} UAV's energy consumption at time slot n, its collective energy consumption in the time range $[n_i, n_f]$ is constrained by the battery capacity $E_{\max}$ as follows:

$$\sum_{n=n_i}^{n_f} E_m^{(n)} \leq E_{\max}. \tag{21}$$

This general constraint is applicable to various scenarios. For instance, in a scenario where UAVs can recharge at charging stations, n_i denotes the time when the m^{th} UAV begins tracking the wildfire and n_f denotes the time it reaches the charging station. Following a recharge, n_i can be reset to initiate a new tracking event.

E. AoI over the Cellular Network

Given the delay-sensitive nature of transmitting wildfire images, it is necessary to keep the AoI at bay to maintain data freshness. However, when UAVs share resources, interference from concurrent transmissions can compromise the AoI. To address this, current network deployments feature N-antenna BSs which can spatially separate the UAV transmissions through multiple-input-multiple-output (MIMO) techniques. To define the AoI, we first need to incorporate certain features, primarily related to the channel model and the calculation of spectral efficiency.

1) Channel Models

The N-dimensional air-to-ground channel between UAV m and the BS features two main components of the line-of-sight (LoS) and non-LoS. Therefore, it follows a Rician distribution where the Rician factor $K_m^{(n)}$ determines which component dominates and is given by

$$K_m^{(n)} = A_1 \exp\left(A_2 \arcsin\left(\frac{h_m^{(n)}}{d_m^{(n)}}\right)\right), \tag{22}$$

where $d_m^{(n)}$ is the distance between the UAV and the BS. Additionally, A_1 and A_2 are environment-dependent parameters [66]. Therefore, the channel vector $\boldsymbol{g}_m^{(n)} \in \mathbb{C}^{N\times 1}$ is

$$\boldsymbol{g}_m^{(n)} = \sqrt{\frac{\beta_0}{\left(d_m^{(n)}\right)^{\kappa}(K_m^{(n)}+1)}}\left[\sqrt{K_m^{(n)}}e^{j\psi_m}\boldsymbol{s}_m^{(n)} + \boldsymbol{a}_m^{(n)}\right], \tag{23}$$

where β_0 is the path loss at a reference distance of 1 meter, κ is the path loss exponent and $\psi_m \sim \mathcal{U}[0, 2\pi]$ reflects drifting. Additionally, $\boldsymbol{a}_m^{(n)} \sim \mathcal{N}_{\mathbb{C}}(\boldsymbol{0}, \boldsymbol{R_a})$ represents the small scale fading for a given spatial correlation matrix $\boldsymbol{R_a}$. It is assumed that the BS features a uniform linear array, and thus the steering vector $\boldsymbol{s}_m^{(n)} \in \mathbb{C}^{N\times 1}$ is

$$[\boldsymbol{s}_m^{(n)}]_n = e^{j\frac{2\pi f_c}{c}d(n-1)\sin(\theta_m^{(n)})\cos(\phi_m^{(n)})}, \tag{24}$$

where $\theta_m^{(n)}$ and $\phi_m^{(n)}$ represent the azimuth and elevation angles between the transmitter and receiver, respectively. The antenna spacing is d, f_c is the operating frequency, and c stands for the speed of light. Therefore, the channel covariance is

$$\begin{aligned}\boldsymbol{R}_m^{(n)} &= \mathbb{E}\{\boldsymbol{g}_m^{(n)}\boldsymbol{g}_m^{(n)*}\} \\ &= \frac{\beta_0}{\left(d_m^{(n)}\right)^{\kappa}(K_m^{(n)}+1)}\left[K_m^{(n)}\boldsymbol{s}_m^{(n)}\boldsymbol{s}_m^{(n)*} + \boldsymbol{R_a}\right].\end{aligned} \tag{25}$$

2) Channel State Information

We consider imperfect channel estimates at the BS, which is often avoided within the UAV literature given the complexity it adds. UAVs are first assigned pairwise orthogonal pilot sequences of length τ and power p. Then, upon pilot reception at the BS, the MMSE estimates follow $\boldsymbol{g}_m^{(n)} = \hat{\boldsymbol{g}}_m^{(n)} + \tilde{\boldsymbol{g}}_m^{(n)}$ where $\hat{\boldsymbol{g}}_m^{(n)}$ is zero-mean with covariance matrix

$$\boldsymbol{\Phi}_m^{(n)} = \mathbb{E}\{\hat{\boldsymbol{g}}_m^{(n)}\hat{\boldsymbol{g}}_m^{(n)*}\} = \boldsymbol{R}_m^{(n)}\boldsymbol{\Psi}_m^{(n)-1}\boldsymbol{R}_m^{(n)}, \tag{26}$$

and $\boldsymbol{\Psi}_m^{(n)} = \boldsymbol{R}_m^{(n)} + \frac{\sigma^2}{\mathrm{p}\tau}\boldsymbol{I}$. The error term $\tilde{\boldsymbol{g}}_m^{(n)}$ is zero-mean with covariance $\boldsymbol{C}_m^{(n)} = \boldsymbol{R}_m^{(n)} - \boldsymbol{\Phi}_m^{(n)}$.

3) Data Transmission

Omitting the time index for simplicity, the signal observed at the BS on a given time-frequency resource is $\boldsymbol{y} = (y_1, \ldots, y_N)^{\mathrm{T}}$ and can be calculated as

$$\begin{aligned}\boldsymbol{y} &= \sum_{m=1}^{M}\boldsymbol{g}_m\sqrt{p_m}s_m + \boldsymbol{n} \\ &= \underbrace{\sum_{m=1}^{M}\hat{\boldsymbol{g}}_m\sqrt{p_m}s_m}_{\text{signals}} + \underbrace{\sum_{m=1}^{M}\tilde{\boldsymbol{g}}_m\sqrt{p_m}s_m + \boldsymbol{n}}_{\text{effective noise: } \boldsymbol{v}},\end{aligned} \tag{27}$$

where s_m is a complex symbol with unit power, p_m is the transmit power, and $\boldsymbol{n} \sim \mathcal{N}_{\mathbb{C}}(\boldsymbol{0}, \sigma^2\boldsymbol{I})$ is the noise. Hence, the effective noise is zero-mean with covariance $\boldsymbol{\Sigma} = \mathbb{E}\{\boldsymbol{v}\boldsymbol{v}^*\} = \sum_{\forall m}\boldsymbol{C}_m p_m + \sigma^2\boldsymbol{I}$. In addition, real deployments might suffer from high interference generated by other networks and/or frequency bands. Also, UAVs may encounter significant signal blockages due to the environment. While these factors are not explicitly taken into account, our formulation is generic enough to consider them. For example, more interference would be equivalent to an increased noise level while blockages would require a higher attenuation in β. Hence, under the model provided previously, and under MMSE reception, optimal from the signal-to-interference-and-noise-ratio (SINR) perspective, the m^{th} UAV experiences a SINR value given by

$$\mathrm{SINR}_m = \hat{\boldsymbol{g}}_m^*\left(\sum_{j\neq m}^{M}\hat{\boldsymbol{g}}_m\hat{\boldsymbol{g}}_m^* p_m + \boldsymbol{\Sigma}\right)^{-1}\hat{\boldsymbol{g}}_m p_m. \tag{28}$$

For a given transmission bandwidth W and pilot overhead $\frac{\tau}{\tau_c}$, the ergodic spectral efficiency is given by:

$$\mathrm{SE}_m^{(n)} = W\delta\left(1 - \frac{\tau}{\tau_c}\right)\mathbb{E}\{\log_2(1 + \mathrm{SINR}_m^{(n)})\}. \tag{29}$$

However, a closed-form expression for Eq. (29) is not available. The numerical evaluations of (29) in optimization problems will cause stability and convergence issues. To address this challenge, random matrix theory poses an interesting

framework, allowing for closed forms that rely exclusively on large-scale parameters.

4) Large-Dimensional Analysis

We evaluate Eq. (29) in the large-scale regime, as $N, M \to \infty$ with finite $\frac{N}{M} > 1$. Convergence to deterministic limits is assured if matrices $\mathbf{\Phi}_m^{(n)}$ have uniformly bounded spectral norms. Omitting the time index, the following result can be deduced.

Theorem 1. *With $N, M \to \infty$ and MMSE reception, $\mathrm{SINR}_m - \overline{\mathrm{SINR}}_m \to 0$ almost surely (a.s.) where*

$$\overline{\mathrm{SINR}}_m = p_m \operatorname{tr}\left[\mathbf{\Phi}_m \left(\sum_{j \neq m}^{K} \frac{\mathbf{\Phi}_j p_j}{1 + e_j} + \mathbf{\Sigma} \right)^{-1}\right]. \tag{30}$$

The coefficients $e_j = \lim_{t\to\infty} e_j^{(t)}$ are iteratively obtained from (71) in Appendix C.

Proof. The proof can be found in Appendix C. □

Restoring the time index and applying the continuous mapping theorem [67], we have $\mathrm{SE}_m^{(n)} - \overline{\mathrm{SE}}_m^{(n)} \to 0$ where

$$\overline{\mathrm{SE}}_m^{(n)} = W\delta \left(1 - \frac{\tau}{\tau_\mathrm{c}}\right) \log_2\left(1 + \overline{\mathrm{SINR}}_m^{(n)}\right), \tag{31}$$

depends solely on large-scale parameters, enhancing stability during the optimization.

5) AoI of the Images

Upon capturing an image, per Eq. (13), the transmission of B bits to the BS is required. To ensure the freshness of data, our metric is the AoI, defined as [47]

$$\eta_m^{(n)} = \begin{cases} \eta_m^{(n-1)} + 1 & \text{if } \Lambda_m^{(n)} = 0 \\ 1 & \text{otherwise} \end{cases}, \tag{32}$$

where $\Lambda_m^{(n)}$ is a binary variable defined as

$$\Lambda_m^{(n)} = \begin{cases} 1 & \text{if } \sum_{i=1}^{n} \overline{\mathrm{SE}}_m^{(i)} \geq nB \\ 0 & \text{otherwise} \end{cases}. \tag{33}$$

In other words, should the BS fail to receive the images, the AoI will continue to increase until all the data gathered by the m^{th} UAV, quantified as nB, is reliably received by the BS. Hence, a constraint on the maximum tolerated AoI arises as

$$\eta_m^{(n)} \leq \eta_{\max} \, \forall m, n, \tag{34}$$

where $\eta_{\max}$ is the maximum AoI allowed by the system. Note that to satisfy (34), it is required to continuously adjust the UAVs' trajectories, including the flying altitudes and transmit powers which affect the $\overline{\mathrm{SINR}}_m^{(n)}$, while satisfying the corresponding constraints introduced in previous subsections.

III. Problem Formulation

This section aims to outline the end-to-end operation for each of the UAVs, combining two major modes within the same optimization framework: (*i*) tracking the wildfire, and (*ii*) charging the batteries as necessary. Therefore, the goal is to find the UAV trajectories, speeds, and transmit powers such that energy-efficient coverage is achieved and the UAVs head to a charging station when their battery levels are low. The switching between tracking and charging modes is controlled by a binary variable $\lambda_m^{(n)}$, which is set to zero when the UAV's energy level falls below a threshold, indicating the need to head to a charging point and one otherwise (refer to Sec. V for further details). Additionally, considering the dependence of $\boldsymbol{q}_m^{(n+1)}$ and $\dot{\boldsymbol{q}}_m^{(n+1)}$ on $\boldsymbol{u}_m^{(n)}$ according to Eq. (1), optimizing with respect to $\boldsymbol{u}_m^{(n)}$ proves to be more convenient. Moreover, a variety of constraints must be satisfied at any given time. These constraints include (*i*) a maximum velocity and acceleration, (*ii*) bounded flying altitudes, (*iii*) ensuring that at every time instant the UAV has enough energy to reach a charging point, (*iv*) the avoidance of collisions, and (*v*) satisfying the AoI and transmit power requirements. Hence, at every time step, the following multi-objective optimization emerges

$$\begin{aligned} &\max_{\boldsymbol{u}_m^{(n)}, p_m^{(n)}} \sum_{m=1}^{M} \left(\lambda_m^{(n)} F_1\big(\boldsymbol{q}_m^{(n)}\big) + \big(1 - \lambda_m^{(n)}\big) F_2\big(\boldsymbol{q}_m^{(n)}\big)\right) \\ &\min_{\boldsymbol{u}_m^{(n)}, p_m^{(n)}} \sum_{m=1}^{M} E_m^{(n)}\left(\boldsymbol{q}_m^{(n)}, \dot{\boldsymbol{q}}_m^{(n)}, p_m^{(n)}\right) \\ &\quad \text{s.t.} \quad (1), (2), (14), (20), (34), \end{aligned} \tag{35}$$

which, can be transformed into a single-objective optimization problem through compromise programming. Hence, by introducing a weighting variable, denoted by μ, the final optimization problem is

$$\begin{aligned} &\max_{\boldsymbol{u}_m^{(n)}, p_m^{(n)}} \sum_{m=1}^{M} \left(\lambda_m^{(n)} F_1\big(\boldsymbol{q}_m^{(n)}\big) + \big(1 - \lambda_m^{(n)}\big) F_2\big(\boldsymbol{q}_m^{(n)}\big)\right) \\ &\qquad - \mu \sum_{m=1}^{M} E_m^{(n)}\left(\boldsymbol{q}_m^{(n)}, \dot{\boldsymbol{q}}_m^{(n)}, p_m^{(n)}\right) \\ &\quad \text{s.t.} \quad (1), (2), (14), (20), (34). \end{aligned} \tag{36}$$

Note that μ is the priority weight that indicates the importance of an objective function, i.e., $\mu \to 0$ reveals that maximizing the first function is more important than minimizing the second one, while $\mu = 1$ assigns the same weight to both. Additionally, $F_1(\cdot)$ is chosen to increase with respect to the coverage and $F_2(\cdot)$ aims at reducing the flying time between the UAV and the nearest charging point. Therefore, $F_2(\cdot)$ is selected to increase as the distance to the charging point decreases. Additionally, to ensure energy-efficient trajectories in each mode, minimizing the required energy $E_m^{(n)}(\cdot)$ is a common objective maintained for all time steps n. Note that since the problem is formulated as a maximization, the energy term is included with a negative sign, scaled by a positive multiplier denoted by μ, effectively minimizing it. Finally, we define $F_1(\cdot)$ and $F_2(\cdot)$ subsequently where the former increases as the coverage increases, and therefore:

$$F_1\big(\boldsymbol{q}_m^{(n)}\big) = \frac{1}{M} C^{(n)}, \tag{37}$$

where $C^{(n)}$ is defined in Eq. (10). Other increasing forms of $C^{(n)}$ could be investigated as well. And, second, $F_2(\cdot)$ is

chosen to be a decreasing function of the distance between the UAV and the closest charging point, i.e.:

$$F_2\big(\boldsymbol{q}_m^{(n)}\big) = \frac{1}{\|\boldsymbol{q}_m^{(n)} - \boldsymbol{q}_k^{\mathrm{c}}\| + \upsilon}, \tag{38}$$

where $k = \arg\min_\ell \|\boldsymbol{q}_m^{(n)} - \boldsymbol{q}_\ell^{\mathrm{c}}\|$, i.e., k is the index of the closest charging point at time n. As per $F_1(\cdot)$, other forms of $F_2(\cdot)$ could be investigated, and υ is a small positive number preventing the ratio to diverge. Since (36) poses a highly non-convex optimization problem, conventional methods such as gradient ascent or SCA are inadequate for finding a solution. Therefore, the next section introduces the essential components required to utilize RL in solving this problem.

IV. Proposed Learning Framework

We first introduce the concept of a multi-agent Markov Decision Process (MDP) and explore the conditions under which a factored MDP can be derived. Following this, we present the definitions required to address the optimization problem in Sec. III using RL.

A. Multi-Agent MDPs

In single-agent MDPs, an agent interacts with the environment by navigating different states and selecting a variety of actions, thereby forming a policy, i.e., a sequence of actions. At each state transition, the environment provides a reward, which accumulates over time as a discounted sum. The primary objective of the agent is to maximize such cumulative reward, serving as a measure of the policy's effectiveness.

In multi-agent MDPs, M agents collectively aim to maximize the cumulative reward received by all agents. Consequently, the action and state spaces are expanded to account for the joint sets. Thus, a multi-agent MDP can be defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, R)$ where the following definitions apply for each of the terms:

- $\mathcal{S}$ represents the state space, which comprises the states of the M agents, denoted as $\mathcal{S} = \{\mathcal{S}_1, \ldots, \mathcal{S}_M\}$. Here, $\boldsymbol{s}^{(n)} = \{\boldsymbol{s}_1^{(n)}, \ldots, \boldsymbol{s}_M^{(n)}\}$ signifies the realization of the state at time n, and should reflect certain features of the optimization problem for the learning to succeed.
- $\mathcal{A}$ is the set of joint actions, i.e., $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_M$. At time n, we represent the action taken by the M agents as $\boldsymbol{a}^{(n)} = \{\boldsymbol{a}_1^{(n)}, \cdots, \boldsymbol{a}_M^{(n)}\}$.
- $\mathcal{P}$ is the transition probability between states after taking a certain action.
- R is the reward function. We denote by $r\left(\boldsymbol{s}^{(n)}, \boldsymbol{a}^{(n)}\right)$ the reward associated to taking action $\boldsymbol{a}^{(n)}$ when in state $\boldsymbol{s}^{(n)}$.

Given that (36) can be represented using multi-agent and single-agent MDP formulations, RL presents an attractive framework for addressing these challenges. Specifically, the aim in RL is to find a policy π that maximizes the expected discounted reward:

$$\pi^* = \arg\max_\pi \mathbb{E}\left\{\sum_{j=0}^{\infty} \gamma^j r\Big(\boldsymbol{s}^{(j)}, \boldsymbol{a}^{(j)}\Big) \,\Big|\, \pi\right\}, \tag{39}$$

where γ is the discount factor and the expectation is over $\boldsymbol{a}^{(n)} \sim \pi\big(\cdot|\boldsymbol{s}^{(n)}\big)$ and $\boldsymbol{s}^{(n+1)} \sim P\big(\boldsymbol{s}^{(n+1)}|\boldsymbol{s}^{(n)}, \boldsymbol{a}^{(n)}\big)$. While directly pursuing the optimal policy is feasible, it is more practical to utilize the Q-function. This function quantifies the expected cumulative reward linked with selecting a particular action $\boldsymbol{a}$ in a given state $\boldsymbol{s}$ and subsequently adhering to a policy π

$$Q^\pi(\boldsymbol{s}, \boldsymbol{a}) = \mathbb{E}_\pi\left\{\sum_{j=0}^{\infty} \gamma^j r\left(\boldsymbol{s}^{(j')}, \boldsymbol{a}^{(j')}\right) \,\Big|\, \boldsymbol{s}^{(n)} = \boldsymbol{s}, \boldsymbol{a}^{(n)} = \boldsymbol{a}\right\}. \tag{40}$$

where $j' = j + n + 1$. In fact, the optimal policy can be found by using the Q-function and the time difference (TD) method, which iteratively updates the Q-values as more interactions with the environment are added [49].

B. Factored MDP

As per [49], employing the TD method on the Q-function theoretically guarantees convergence to the optimal policy. However, note that the previous subsection considers the joint set of states and actions for M agents. This poses a significant challenge because the state and action spaces expand exponentially with the number of agents, rendering the problem intractable. However, a large multi-agent MDP can be factored into single-agent MDPs when the interactions between agents are limited or can be approximated locally. This factorization simplifies the problem by breaking it down into smaller and more manageable parts, each involving a single agent [50], [51]. This simplification also allows for parallelization, leading to faster convergence and more efficient learning. Therefore, under the assumption that the reward function can be factored into M individual functions

$$r\left(\boldsymbol{s}^{(n)}, \boldsymbol{a}^{(n)}\right) = \sum_{m=1}^{M} r_m\left(\boldsymbol{s}_m^{(n)}, \boldsymbol{a}_m^{(n)}\right), \tag{41}$$

it can be easily shown that the Q-function also factorizes into M disjoint functions, each following a different policy π_m:

$$Q^\pi(\boldsymbol{s}, \boldsymbol{a}) = \sum_{m=1}^{M} Q^{\pi_m}(\boldsymbol{s}_m, \boldsymbol{a}_m). \tag{42}$$

C. TD3 Algorithm

Based on [49], learning the Q-function in small and discrete state and action spaces is feasible. However, problems such as (36), featuring continuous domains, demand innovative solutions. To address this, DQL offers an interesting framework where the Q-function is represented by a neural network with parameters $\boldsymbol{\theta}$, i.e., $Q(\boldsymbol{s}, \boldsymbol{a}) \approx Q(\boldsymbol{s}, \boldsymbol{a}; \boldsymbol{\theta})$. For brevity, this subsection omits the subscript m although the derivations are conducted for each agent since we employ the factored MDP approach. Based on the target value obtained through:

$$y(\boldsymbol{s}, \boldsymbol{a}) = r(\boldsymbol{s}, \boldsymbol{a}) + \gamma \max_{\boldsymbol{a}'} Q(\boldsymbol{s}', \boldsymbol{a}'; \boldsymbol{\theta}'), \tag{43}$$

the network parameters are adjusted to minimize

$$L(\boldsymbol{\theta}) = \mathbb{E}\big\{|y(\boldsymbol{s}, \boldsymbol{a}) - Q(\boldsymbol{s}, \boldsymbol{a}; \boldsymbol{\theta})|^2\big\}. \tag{44}$$

where $\boldsymbol{\theta}^{'}$ represent the target network parameters, which remain unchanged over F updates and keep $y(\boldsymbol{s},\boldsymbol{a})$ fixed over multiple updates. Although $Q(\boldsymbol{s},\boldsymbol{a};\boldsymbol{\theta})$ can handle continuous states, further modifications are required to accommodate continuous actions. To circumvent this challenge, we employ policy-based algorithms, particularly the TD3, an improved version of the well-known deep deterministic policy gradient (DDPG) [57]. Under the DDPG approach, the policy π is represented by a neural network with parameters $\boldsymbol{\phi}$, i.e., $\pi_{\boldsymbol{\phi}}$, updated through

$$\nabla J(\boldsymbol{\phi}) = \mathbb{E}\big\{\nabla_{\boldsymbol{\phi}}\pi_{\boldsymbol{\phi}}(\boldsymbol{s})\ \nabla_{\boldsymbol{a}}Q(\boldsymbol{s},\boldsymbol{a};\boldsymbol{\theta})|_{\boldsymbol{a}=\pi_{\boldsymbol{\phi}}(\boldsymbol{s})}\big\}. \tag{45}$$

Finally, TD3 introduces the following three improvements that enhance stability during the learning over DDPG [56]. First, updates to the policy parameters occur less frequently compared to the Q-function parameters, decreasing the accumulation of residual errors [57]. Second, TD3 includes two twin blocks aiming at learning two different Q-functions with parameters $\boldsymbol{\phi}_1$ and $\boldsymbol{\phi}_2$. Consequently, (43) is modified accordingly as

$$y(\boldsymbol{s},\boldsymbol{a}) = r(\boldsymbol{s},\boldsymbol{a}) + \gamma \min_{i=1,2} Q(\boldsymbol{s}',a';\boldsymbol{\theta}_i'). \tag{46}$$

And, third, TD3 incorporates noise into the target action as

$$\hat{\boldsymbol{a}} = \pi_{\boldsymbol{\phi}'}(\boldsymbol{s}) + \hat{\epsilon}, \tag{47}$$

where $\hat{\epsilon} \sim \text{clip}\big(\mathcal{N}(0,\hat{\sigma}_a), -\hat{\epsilon}_{\max}, \hat{\epsilon}_{\max}\big)$, i.e., given an interval $[-\hat{\epsilon}_{\max}, \hat{\epsilon}_{\max}]$, the clip-function limits the value of the input to the interval boundaries.

Next, we define the states, actions, and rewards for each agent such that (36) can be solved using TD3 via factored MDPs.

D. Multi-Agent MDP for Wildfire Coverage

To successfully train a TD3-based agent for wildfire coverage, we need to define states, actions, and rewards.

1) States

The states $\boldsymbol{s}_m^{(n)}$ inform about the UAV locations and its dynamics, the wildfire location, charging stations, the AoI, and the inter-UAV distances. To be precise, the first three elements of the states relate to the 3D UAV locations as

$$\boldsymbol{e}_m^{(n)} = \left\{\frac{x_m^{(n)}}{S}, \frac{y_m^{(n)}}{S}, \frac{h_m^{(n)} - h_{\min}}{h_{\max} - h_{\min}}\right\}. \tag{48}$$

where S is a normalization constant. To satisfy the maximum speed and acceleration constraints, the normalized speeds from the previous time slot are also taken into account:

$$\boldsymbol{w}_m^{(n)} = \left\{\frac{\dot{x}_m^{(n-1)}}{\dot{q}_m^{\max}}, \frac{\dot{y}_m^{(n-1)}}{\dot{q}_m^{\max}}, \frac{\dot{h}_m^{(n-1)}}{\dot{q}_m^{\max}}\right\}. \tag{49}$$

Next, we incorporate information regarding the remaining uncovered fire perimeter. Since this is fundamentally a tracking problem, we compute the uncovered density's center of mass as {

$$\begin{aligned}&\big(\mathrm{x}_{\mathrm{c}}^{(n)}, \mathrm{y}_{\mathrm{c}}^{(n)}\big) = \\ &\frac{\int_{\mathcal{F}\setminus \bigcup_{m=1}^{M}\lambda_m^{(n)}\mathcal{B}_m^{(n)}} xy\mathbb{I}\{\psi(x,y,n)>0\}\,dxdy}{\int_{\mathcal{F}\setminus \bigcup_{m=1}^{M}\lambda_m^{(n)}\mathcal{B}_m^{(n)}} \mathbb{I}\{\psi(x,y,n)>0\}\,dxdy}.\end{aligned} \tag{50}$$

Note that the integration region consists of what remains outside the tracking UAVs' FoV, i.e., $\mathcal{F} - \bigcup_{m=1}^{M}\lambda_m^{(n)}\mathcal{B}_m^{(n)}$. Therefore, by considering the non-zero density points, we can compute the uncovered fire's mass center. Consequently, the next two states are given by

$$\boldsymbol{v}_m^{(n)} = \begin{cases} \left\{\frac{\mathrm{x}_{\mathrm{c}}^{(n)}}{S}, \frac{\mathrm{y}_{\mathrm{c}}^{(n)}}{S}\right\} & \text{if } c^{(n)} < 1 \text{ and } \lambda_m^{(n)} = 1 \\ \left\{\frac{x_k^c}{S}, \frac{y_k^c}{S}\right\} & \text{if } \lambda_m^{(n)} = 0 \\ \left\{\frac{x_m^{(n)}}{S}, \frac{y_m^{(n)}}{S}\right\} & \text{otherwise} \end{cases}. \tag{51}$$

In other words, $\boldsymbol{v}_m^{(n)}$ provides information about the desired locations for the next move, which could be the uncovered fire's mass center, the closest charging point or the current location. Additionally, we include the inter-UAV distances $d_{m,j}^{(n)}, m \neq j$, as part of the state space

$$\ell_{m,j}^{(n)} = \begin{cases} 1 & \text{if } d_{m,j}^{(n)} \leq D_{\text{safe}} \\ \frac{d_{\text{safe}} - d_{m,j}^{(n)}}{d_{\text{safe}} - D_{\text{safe}}} & \text{if } d_{\text{safe}} > d_{m,j}^{(n)} > D_{\text{safe}} \\ 0 & \text{if } d_{m,j}^{(n)} \geq d_{\text{safe}} \end{cases}, \tag{52}$$

where $d_{\text{safe}} > D_{\text{safe}}$ is a large enough distance for which no action is needed. Hence, $\boldsymbol{\ell}_m^{(n)} = \{\ell_{m,j}^{(n)}\ \forall j \neq m\}$. Finally, the AoI is included to represent UAVs' communication constraints. Altogether, the state space is $(M+8)$-dimensional and comprises:

$$\boldsymbol{s}_m^{(n)} = \left\{\boldsymbol{e}_m^{(n)}, \boldsymbol{w}_m^{(n)}, \boldsymbol{v}_m^{(n)}, \boldsymbol{\ell}_m^{(n)}, \eta_m^{(n)} - 1\right\}. \tag{53}$$

2) Actions

The action $\boldsymbol{a}_m^{(n)}$ updates the UAV trajectories and transmit power. To be precise, $\boldsymbol{a}_m^{(n)} \in \mathbb{R}^{4\times 1}$ where the first three coordinates are the position reference signal $\boldsymbol{u}_m^{(n)}$ in Eq. (1) and the fourth component is the transmit power $p_m^{(n)}$. Therefore, $\boldsymbol{a}_m^{(n)} = \left(\boldsymbol{u}_m^{(n)}, p_m^{(n)}\right)$.

3) Rewards

The reward function defines the immediate feedback the agent receives from the environment after taking a certain action, guiding the learning process. Such a function contains information about the cost function and the constraints, and upon following the reward shaping technique, it can be defined as [68]:

$$r\left(\boldsymbol{s}_m^{(n)}, \boldsymbol{a}_m^{(n)}\right) = \sum_{j=1}^{6} r_j\left(\boldsymbol{s}_m^{(n)}, \boldsymbol{a}_m^{(n)}\right), \tag{54}$$

where each of the contributions is described subsequently. The first term relates to the coverage that the swarm achieves if

$\lambda_m^{(n)} = 1$ or how closer the UAV gets to a charging point if $\lambda_m^{(n)} = 0$. It is presented in (55), on top of the next page, where $C^{(n+1)}$ follows Eq. (10) with the updated FoV for the m^{th} UAV through $\mathcal{B}_m^{(n+1)}$ and $K_{\text{c}} > 0$ is a constant. Next, the agent incurs a penalty when the maximum velocity is exceeded, given by:

$$r_2\Big(\boldsymbol{s}_m^{(n)}, \boldsymbol{a}_m^{(n)}\Big) = \begin{cases} -K_{\text{vel}} & \text{if } \|\dot{\boldsymbol{q}}_m^{(n)}\| > \dot{q}_m^{\max} \\ 0 & \text{otherwise} \end{cases}, \tag{56}$$

for $K_{\text{vel}} > 0$. Analogously, for the acceleration:

$$r_3\Big(\boldsymbol{s}_m^{(n)}, \boldsymbol{a}_m^{(n)}\Big) = \begin{cases} -K_{\text{acc}} & \text{if } \|\Delta\dot{\boldsymbol{q}}_m^{(n)}\| > \ddot{q}_m^{\max}\delta \\ 0 & \text{otherwise} \end{cases}. \tag{57}$$

for $K_{\text{acc}} > 0$. A similar reward prevents the agent from flying out of limits with $r_4(\cdot)$, with a penalty of $K_{\text{lim}} > 0$. Also, when the agent fails to meet the AoI constraint, a penalty of K_{aoi} is observed in $r_5(\cdot)$. Finally, collision avoidance is key and therefore the following reward is included to prevent it:

$$r_6\Big(\boldsymbol{s}_m^{(n)}, \boldsymbol{a}_m^{(n)}\Big) = \begin{cases} -K_{\text{coll}} & \text{if } \exists\ d_{m,j}^{(n)} < D_{\text{safe}} \\ -K_{\text{coll}}/20 & \text{if } \exists\ d_{\text{safe}} > d_{m,j}^{(n)} \\ & \text{and } d_{m,j}^{(n)} > D_{\text{safe}} \\ 0 & \text{otherwise} \end{cases}. \tag{58}$$

In other words, when a collision occurs, i.e., $d_{m,j}^{(n)} < D_{\text{safe}}$, a negative reward of K_{coll} is observed. Additionally, if the UAV approaches within a certain proximity of other UAVs, a smaller penalty is applied to prevent it from getting any closer.

With the above definitions, the TD3 algorithm can be employed to solve (36) as outlined in Alg. 1. Since all UAVs share the same objective, training can be performed using a single agent and the resulting model can then be distributed to other agents, significantly reducing computational costs. This strategy is viable because training the model across different agents yields identical neural network parameters due to the common objectives. Additionally, note that to initialize Alg. 1, the number of independent fire realizations N_{e} is required. Furthermore, the memory replay buffer has a size of $|\mathcal{M}|$ and stores transitions of the form $\Big\{\boldsymbol{s}_m^{(n)}, \boldsymbol{a}_m^{(n)}, r\Big(\boldsymbol{s}_m^{(n)}, \boldsymbol{a}_m^{(n)}\Big), \boldsymbol{s}_m^{(n+1)}\Big\}$ while the network parameters are typically initialized randomly. Finally, each episode can be terminated for various reasons, listed as follows: collision occurrence, UAV exceeding maximum velocity, UAV not satisfying acceleration constraint, UAV flying out of bounds, or achieving a maximum number of time steps.

The computational complexity of Alg. 1 is primarily driven by the training process. Since the feedforward and backpropagation algorithms exhibit the same complexity, it is sufficient to analyze one of them. In this study, both the actor and critic networks consist of three fully connected layers, each with 256 neurons. The input layer of the actor has a dimension of $l_a = M+8$, while the input layer of the critic has a dimension of $l_c = l_a+4$, as both states and actions are required as inputs. The process of adjusting the neuron weights mainly involves matrix multiplications followed by the application of activation functions. Assuming a batch size of N_{mem} training samples and two matrices with dimensions (i, j) and (j, k), the worst-case time complexity for the first layer is $\mathcal{O}(i\times j\times k\times N_{\text{mem}})$. After applying the activation function, the total complexity becomes $\mathcal{O}(i\times j\times k\times N_{\text{mem}}+k\times N_{\text{mem}}) = \mathcal{O}(i\times j\times k\times N_{\text{mem}})$. Thus, based on the sizes of our neural networks, i.e., 256 neurons per layer, it can be shown that the training complexity of the actor is $\mathcal{O}_a\Big(N_{\text{e}}\times l_a\times 256^2\Big)$. A similar analysis applies to the critic, and we denote the corresponding complexity as $\mathcal{O}_c$. Therefore, the overall complexity of Alg. 1 is $\mathcal{O}_a+2\times\mathcal{O}_c$, with the factor of 2 arising from the training of two critic networks.

Algorithm 1: TD3 Algorithm for (36)

Input: No. of episodes N_{e}, memory replay buffer $\mathcal{M}$ and network parameters $\boldsymbol{\phi}$, $\boldsymbol{\theta}_1$, $\boldsymbol{\theta}_2$.
Set $\boldsymbol{\theta}_1^{'} \leftarrow \boldsymbol{\theta}_1$, $\boldsymbol{\theta}_2^{'} \leftarrow \boldsymbol{\theta}_2$, and $\boldsymbol{\phi}^{'} \leftarrow \boldsymbol{\phi}$,
for $e = 1, \ldots, \text{E}$ **do**
  Set $n = 0$ and initialize UAV, BS and wildfire.
  while *not done* **do**
    Select $\boldsymbol{a}_m^{(n)} = \text{clip}\Big(\pi_{\boldsymbol{\phi}}\big(\boldsymbol{s}_m^{(n)}\big) + \epsilon\Big)$, with $\epsilon \sim \mathcal{N}(0, \sigma_a)$, and observe $\boldsymbol{s}_m^{(n+1)}$.
    Calculate the reward as per (54).
    Store $\Big\{\boldsymbol{s}_m^{(n)}, \boldsymbol{a}_m^{(n)}, r\Big(\boldsymbol{s}_m^{(n)}, \boldsymbol{a}_m^{(n)}\Big), \boldsymbol{s}_m^{(n+1)}\Big\}$ in $\mathcal{M}$.
    Randomly choose N_{mem} experiences from $\mathcal{M}$.
    Obtain target actions $\hat{\boldsymbol{a}}_m^{(n)}$ based on (47).
    Calculate target value $y(s, \boldsymbol{a})$ as per (43).
    Update critic networks by minimizing (44).
    if n *mod* 2 **then**
      Use (45) to update $\boldsymbol{\phi}$.
      Update target networks:
      $\boldsymbol{\theta}^{'} \leftarrow (1-\tau_{\text{T}})\boldsymbol{\theta}^{'} + \tau_{\text{T}}\boldsymbol{\theta}$
      $\boldsymbol{\phi}_i^{'} \leftarrow (1-\tau_{\text{T}})\boldsymbol{\phi}_i^{'} + \tau_{\text{T}}\boldsymbol{\phi}_i$ for $i = 1, 2$.
    end
    Grow wildfire.
    $n = n + 1$.
  end
end

V. Numerical Results

To assess the proposed solution's performance, a 250-m×250-m simulation environment with the parameters in Table II is considered. With the goal of emulating real wildfire scenarios, the parameters for fire propagation are selected from [3], [8], [9]. The UAVs are assumed to have $\xi = 6$ propellers and be able to hover at 40% throttle. In addition, UAVs are assumed to carry real cameras and therefore the half-view angles $\boldsymbol{\alpha}$ are set accordingly [69], while the UAV and channel parameters are borrowed from [26], [70]. Moreover, following the suggestions from [43]–[46], we set the TD3 parameters, where the variance of the noise added to the actions is $\sigma_a = \hat{\sigma}_a = 0.1$, $\hat{\epsilon}_{\max} = 0.5$, and the parameter soft updates are handled with $\tau_{\text{T}} = 0.01$. The normalization factors for the states and rewards are determined through cross-validation and presented in Table III; alternative values may also be effective. Finally, the UAV, BS, charging point, and ignition locations are initialized at random whereas at each time slot, the wildfire perimeter grows according to FARSITE model

$$r_1\left(\boldsymbol{s}_m^{(n)}, \boldsymbol{a}_m^{(n)}\right) = \lambda_m^{(n)} K_{\mathrm{c}} C^{(n+1)} - K_{\mathrm{e}} E_m^{(n)} + \left(1-\lambda_m^{(n)}\right) \begin{cases} K_{\mathrm{fin}} & \text{if } \boldsymbol{q}_m^{(n)} = \boldsymbol{q}_k^{\mathrm{c}} \\ K_{\mathrm{d}}\left(\|\boldsymbol{q}_k^{\mathrm{c}} - \boldsymbol{q}_m^{(n)}\| - \|\boldsymbol{q}_k^{\mathrm{c}} - \boldsymbol{q}_m^{(n+1)}\|\right) & \text{otherwise} \end{cases}. \tag{55}$$

TABLE II: Simulation parameters

Description	Parameter	Value	Description	Parameter	Value
Camera half-view angles	α_1, α_2	17.5°, 13.125°	Maximum power	$p_{\max}$	100 mW
No. of charging points	C	4	Pathloss at 1 m	β_0	-30 dB
Compression factor	ρ	0.4	Noise power	σ^2	-96 dBm
No. of antennas	N	20	Dense urban param.	A_1, A_2	0, 6.4 dB
Carrier frequency	f_{c}	2.4 GHz	Time slot	δ	0.5 s
Maximum acceleration	$\bar{q}_m^{\max}$	1	Transmission bandwidth	B	10 MHz
Maximum speed	$\dot{q}_m^{\max}$	20	Learning rate	γ	0.85
Minimum and maximum altitude	$h_{\min}$, $h_{\max}$	125 m, 150 m	Maximum AoI	$\eta_{\max}$	4
Pilot sequence length	τ	200	No. of channel uses	τ_{c}	6250
Mean midflame wind speed	U_0	5 m/s	Safety distances	$D_{\mathrm{safe}}, d_{\mathrm{safe}}$	4, 8
Midflame wind speed variance	σ_U	1 m/s	Mean wind direction	$\hat{\theta}_{\mathrm{wind}}$	$\mathcal{U}[0, 2\pi]$
Fire spreading rate	R	35 m/min	Wind direction variance	σ_{wind}	0.1
UAV mass	m	3.49 kg	Length of propellers	r	21 cm
Lift coefficient	$\mathsf{C_L}$	0.0435	Drag coefficient	$\mathsf{C_D}$	0.002
Pitching Moment coefficient	$\mathsf{C_M}$	0.00016	Geometric pitch	J	0.5

in Sec. II-B. While the setup might seem constrained given the dimensions of the region, the number of UAVs used, i.e., $M = 1, \ldots, 4$, and their specification (small energy capacity and limited sensing regions), the simulation environment of 250×250 m^2 is in fact a large region for chosen UAVs. Scaling the simulation environment to a much larger region requires larger UAVs with higher energy capacity and sensing regions. However, the information needed to model the energy consumption and dynamics of more capable UAVs are not available. Given the limited knowledge in this field and the lack of results, this work serves as a first study to gauge the feasibility of considering all aspects of the problem together.

Next, we proceed to optimize the UAV trajectories. Initially considering unlimited energy, Fig. 3 illustrates the average training reward against the number of episodes for $M = 1, \ldots, 4$ for the TD3 and DDPG algorithms, respectively. During the first 1,000 episodes, the replay buffer stores system transitions for both algorithms[2]. Hence, training starts after the one-thousand-episode mark. First, note that there is a clear difference between the two algorithms, i.e., the DDPG baseline fails to learn which actions are beneficial for all values of M. However, the TD3 algorithm clearly improves the average reward after the one-thousand episode mark. In fact, in the initial episodes, the UAV movements result in highly negative rewards, i.e., multiple constraints are not satisfied and the coverage is poor. However, as the number of episodes increases, the average reward increases as well. In fact, the reward stabilizes after 5,000 episodes. Therefore, for subsequent evaluations, we utilize the models stored at the 8,000 episode mark. Finally, note that the same average reward for different values of M does not necessarily mean that the coverage achieved by different numbers of UAVs will be the same. Given the distributed nature of training, similar rewards are expected for different values of M. However, coverage is calculated for the entire swarm, resulting in a larger coverage for higher values of M.

[2]Thus, all movements are random and adding UAVs might not help during the first 1,000 episodes.

TABLE III: State and reward parameters

Description	**Parameter**	**Value**
Normalization in (48), (51)	S	250
Constants in (55)	$K_{\mathrm{c}}, K_{\mathrm{e}}, K_{\mathrm{fin}}, K_{\mathrm{d}}$	50, 0.1, 200, 0.5
Penalty in (56)	K_{vel}	70
Penalty in (57)	K_{acc}	80
Penalty in (58)	K_{coll}	100
Flying-out-of-bounds penalty	K_{lim}	60
Out-of-energy penalty	K_{en}	100

Fig. 4 investigates the coverage dependency on the number of UAVs and their flying altitudes. To isolate the importance of other parameters, we still consider unlimited UAV energy. With the aim of providing general results, we measure the coverage over 1,000 independent wildfire realizations and average such measurements over each time instant. Precisely, Fig. 4a considers $h_{\min} = 125$ m and $h_{\max} = 150$ m, whereas Fig. 4b considers $h_{\min} = 100$ m and $h_{\max} = 125$ m. Clearly, reducing the flying altitudes degrades the coverage. In addition, when the fire grows extensive, e.g., $n > 300$, swarms with just $M = 1$ and $M = 2$ suffer a degradation compared to higher values of M given that low values of M

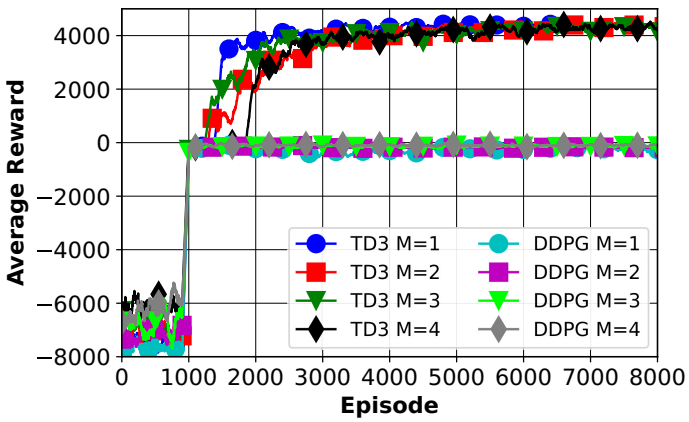
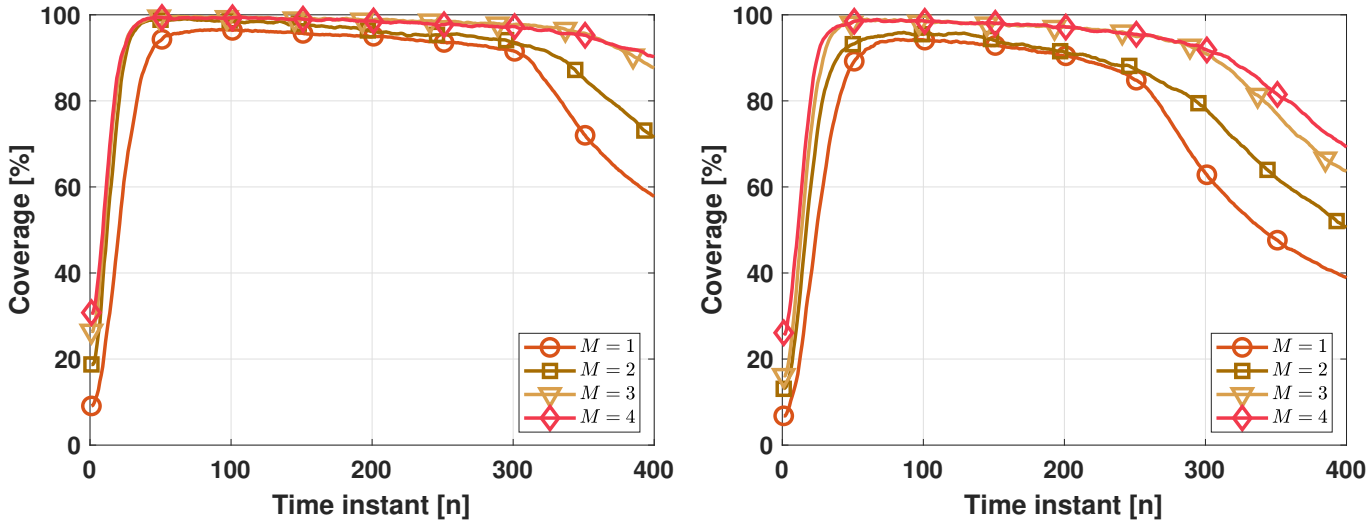
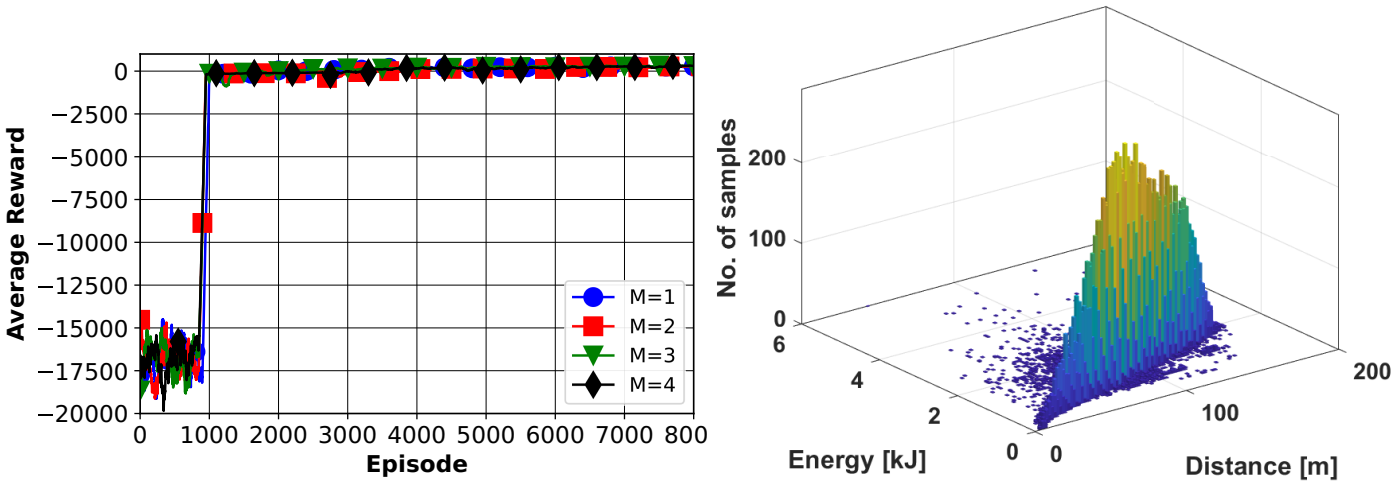


Fig. 3: Average training reward for the tracking for different M.



(a) (b)

Fig. 4: Coverage for tracking and $(h_{\min}, h_{\max})$: (a) (125,150) (b) (100,125).



(a) (b)

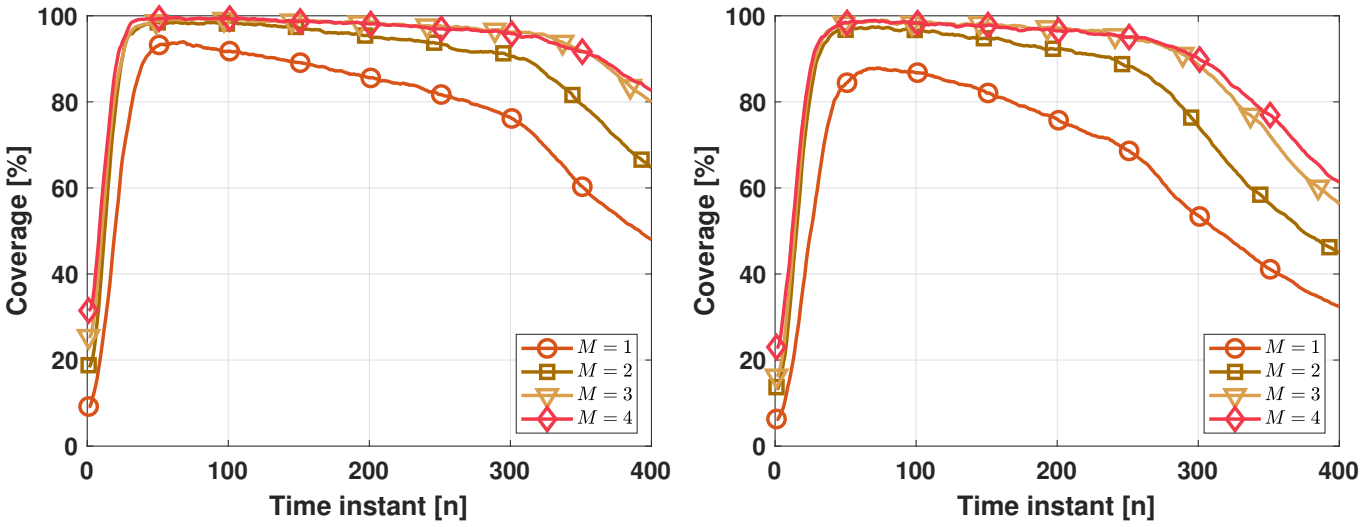
Fig. 5: (a) average training reward when $\lambda_m^{(n)} = 0$; (b) 2D histogram for the distance vs energy in a trained agent for $M = 4$.

are not enough to cover a larger area. In fact, depending on the value of M, the degradation is between 20-30%. Additionally, a relatively small swarm, e.g. $M = 3$ or $M = 4$, can reliably provide a coverage well above 85% during the entire mission for $h_{\min} = 125$ m and $h_{\max} = 150$ m.

To consider the full end-to-end system, first, the value of $\lambda_m^{(n)}$ needs to be determined. This binary variable dictates whether the UAV focuses on wildfire coverage ($\lambda_m^{(n)} = 1$) or heads to a charging point ($\lambda_m^{(n)} = 0$). Thus, solving (36) for $\lambda_m^{(n)} = 0$ enables estimating the required energy and time required for a UAV to reach a charging point, which will dictate the value of $\lambda_m^{(n)}$. The average training reward of this subproblem is depicted in Fig. 5a for different M. Clearly, after the 3,000 episode mark, the models are fully trained and achieve a reward near 200, i.e., the UAV has reached the charging point. Then, we evaluate the trained models over 50,000 realizations and plot a 2D histogram in Fig. 5b with two axes: (i) required energy to reach a charging point, and (ii) initial distance from the UAV to such charging point. To ensure that the UAV reliably reaches the charging point, a conservative threshold must be selected. In this work, we set $\lambda_m^{(n)} = 0$ when the UAV's energy level drops below 1.2 times the average energy at that specific distance plus 1 kJ; other strategies would work as well.

Next, finite energy batteries are considered. Precisely, Fig. 6 assumes $E_{\max} = 125$ kJ, where (a) is for $h_{\min} = 125$ m and $h_{\max} = 150$ m; while (b) is for $h_{\min} = 100$ m and $\{h_{\max} = 125$ m. A similar conclusion to that drawn from Fig. 4 emerges: decreasing the altitude diminishes coverage as the fire expands. Additionally, note that for $M = 1$ and $M = 2$, finite energy batteries yield a 10-15% degradation by the mission's end compared to the scenario with unlimited energy. Conversely, higher values of M can maintain coverage similar to that achieved with unlimited energy batteries.

Next, we analyzed the sensitivity of our RL solution to the



(a) (b)

Fig. 6: Coverage for $E_{\max} = 125$ kJ and $(h_{\min}, h_{\max})$: (a) (125,150) (b) (100,125).

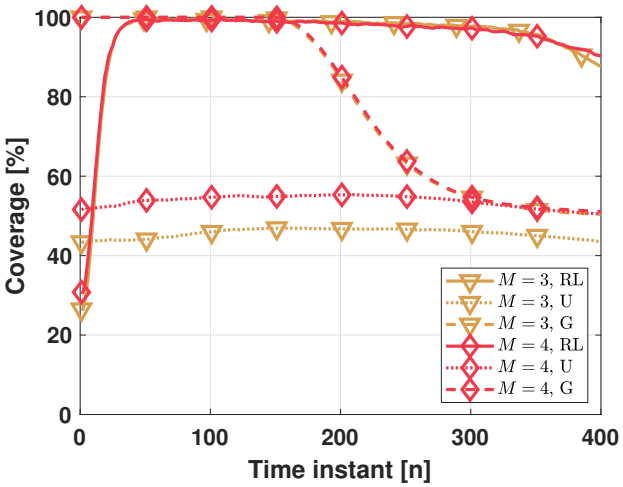


Fig. 7: Coverage for $M = 3$ and $M = 4$ under the proposed RL solution, U-deployment and G-deployment.

initial using two benchmarks that consider static UAV deployments. The first benchmark, labeled U, involves randomly and uniformly distributing the UAVs across the region. The second benchmark, denoted G, positions the UAVs near the ignition point based on a 2D Gaussian random distribution with a covariance of $10\boldsymbol{I}$. Figure 7 shows the results for $M = 3$ and $M = 4$, with $H_{\min} = 125$ m and $H_{\max} = 150$ m. The following observations are made:

1) The RL-based approach provides the highest average coverage throughout the entire mission, for both $M = 3$ and $M = 4$.
2) The G benchmark performs effectively in the early stages. However, as the fire spreads, the coverage declines sharply. In contrast, our approach maintains strong coverage for both initializations.
3) The U method provides the worst performance, provided that the UAV deployment is independent of the fire location.

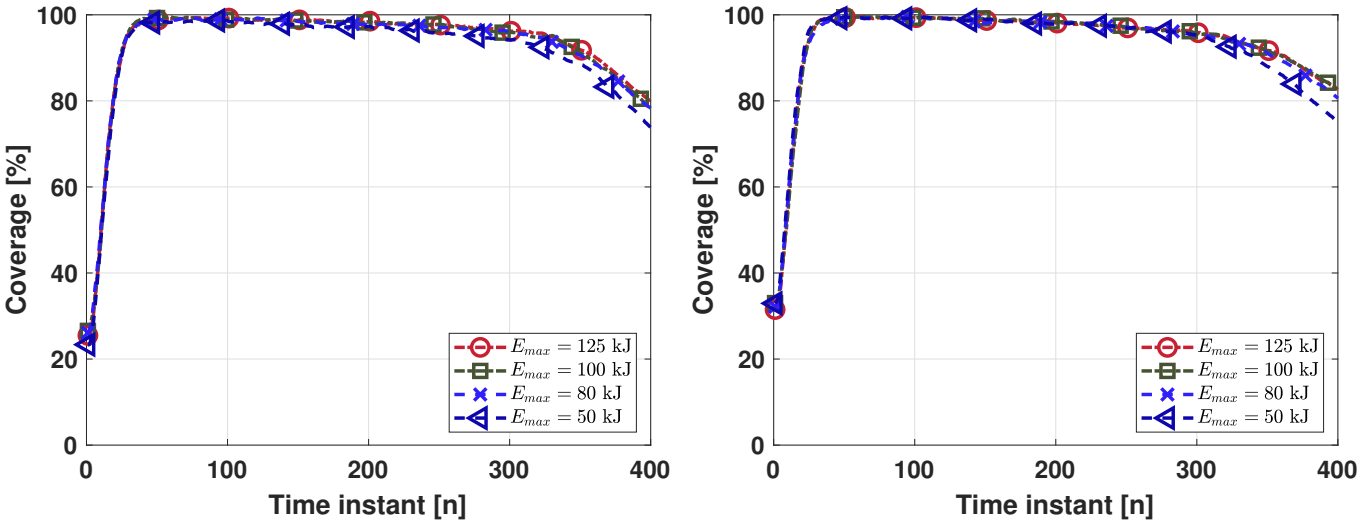
A. *Optimization trade-offs*

In the proposed framework, performance is influenced by several key critical parameters. To study the sensitivity of performance to these system parameters, we investigated the impact of parameters such as the number of UAVs, UAVs' energy capacity, objective weighting variable μ, and wildfire spreading rate R. As discussed before, Figs. 3 and 4 investigate the coverage dependency on the number of UAVs and their flying altitudes. As shown in the figures, while a higher number of UAVs improves coverage, the improvement is minimal for normal fire growth but more noticeable during extensive fire growth. Obviously, training with more UAVs takes longer, as expected.

Next, the relationship between coverage and the energy level of UAV batteries is examined in Fig. 8 for $h_{\min} = 150$ m and $h_{\max} = 125$ m. Specifically, Fig. 8a depicts the average coverage over n for $M = 3$, while Fig. 8b illustrates the same for $M = 4$. Clearly, in both cases, if $E_{\max} \geq 80$kJ, the degradation with respect to much higher energy levels, such as $E_{\max} = 125$kJ, is basically null. Reducing $E_{\max}$ to 50 kJ leads to a noticeable degradation, evident as the wildfire grows large, such as for $n = 400$, where coverage decreases by approximately 8% and 4% for $M = 3$ and $M = 4$, respectively. However, maintaining moderate energy levels, such as $E_{\max} = 80$ kJ, ensures coverage well above 90% for most of the mission.

Additionally, the impact of different values of μ is studied in Fig. 9 for $M = 3$ with $H_{\min} = 125$ m and $H_{\max} = 150$ m. To be precise, the models are trained and evaluated under different values of the Lagrangian multiplier in Eq. (36). Clearly, small values produce similar coverage while highly increasing μ, e.g., for 10^3, the coverage is drastically reduced given that the UAV objective is primarily driven by energy minimization, not coverage maximization.

Finally, Fig. 10 shows, for $M = 3$, the coverage for



(a) (b)

Fig. 8: Coverage for different $E_{\max}$ (a) $M = 3$, (b) $M = 4$.

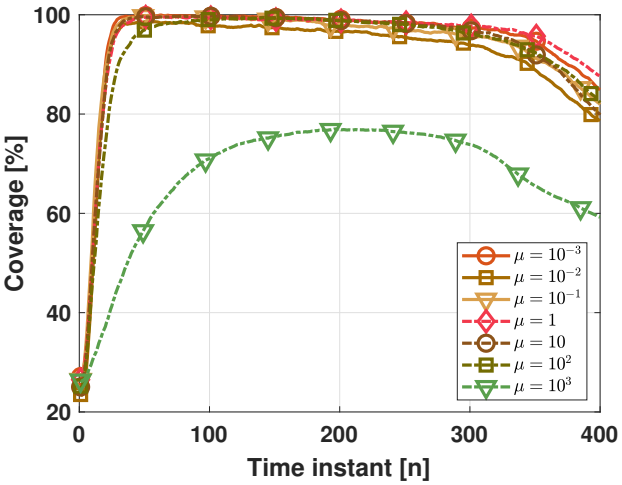


Fig. 9: Coverage for $M = 3$, $H_{\min} = 125$ m and $H_{\max} = 150$ m for different μ values.

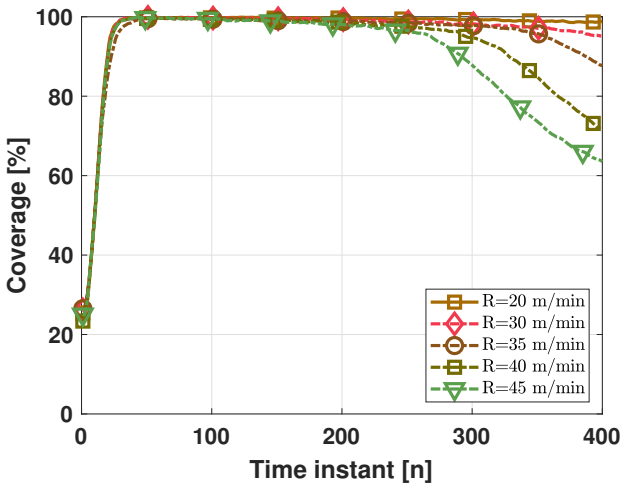


Fig. 10: Coverage for $M = 3$ for different spreading rates R.

different wildfire spreading rates R [m/min]. Slower spreading rates result in larger coverage throughout the entire mission, achieving a value above 95% for R = 20 and R = 30 at all times. However, increasing the spreading rate degrades performance. This is because a higher spreading rate increases the size of the perimeter more quickly, and as a result, for the same number of UAVs, the coverage is compromised when R is large.

VI. Conclusion

This manuscripts has investigated the use of UAVs for wildfire coverage under a variety of dynamical, communications, and energy constraints. Particularly, the goal is to generate energy-efficient UAV trajectories that maximize the wildfire coverage while satisfying the aforementioned constraints. Reinforcement learning tools, and more particularly the TD3 algorithm, have been used to solve such a challenging problem. Our results show that a small swarm, e.g. $M = 4$ UAVs, can consistently provide a high coverage at standard altitudes while using moderate batteries in terms of energy storage. Concretely, a coverage well above 85% can be achieved with a small number of UAVs, although the value is smaller for lower altitudes and energy levels. The dependency between the swarm size, the flying altitudes, and the energy capacity has been studied.

Appendix A

The minimum required power for a UAV to hover can be calculated using dimensional analysis [71]. Let m be the UAV's mass, g the gravitational acceleration magnitude, and L the lift force magnitude produced by its ξ propellers. Assuming uniform hovering flight, via Newton's second law:

$$\xi \mathsf{L} = \mathsf{mg}. \tag{59}$$

According to [71], [72] and using (59), the required velocity v for the tip of each propeller to generate lift is

$$0.5\xi\zeta \mathsf{v}^2\pi \mathsf{r}^2\mathsf{C_L} = \mathsf{mg} \quad \Rightarrow \mathsf{v} = \sqrt{\frac{2\mathsf{mg}}{\xi\zeta\pi\mathsf{r}^2\mathsf{C_L}}}, \tag{60}$$

where r is the length of each propeller. This tip velocity is only achieved when the propellers overcome the drag force. Denoting the drag force magnitude by D, the drag-to-lift ratio $\frac{\mathsf{D}}{\mathsf{L}}$ for a propeller is constant and equals to $\frac{\mathsf{C_D}}{\mathsf{C_L}}$ (see [71] for more information). Thus, the minimum required power P (which is force $\times$ velocity in propellers' tip) is calculated as

$$\xi\mathsf{L} = \mathsf{mg} \xrightarrow{\times \frac{\mathsf{D}}{\mathsf{L}}} \xi\mathsf{L}\frac{\mathsf{D}}{\mathsf{L}} = \mathsf{mg}\frac{\mathsf{D}}{\mathsf{L}} \xrightarrow{\times \mathsf{v}} \xi\mathsf{Dv} = \mathsf{mg}\frac{\mathsf{C_D}}{\mathsf{C_L}}\mathsf{v}$$

$$P = \xi\mathsf{Dv} = \mathsf{mg}\frac{\mathsf{C_D}}{\mathsf{C_L}}\sqrt{\frac{2\mathsf{mg}}{\xi\zeta\pi\mathsf{r}^2\mathsf{C_L}}}. \tag{61}$$

Appendix B

The minimum power required for a UAV to move while ignoring thrust vector adjustment maneuvers can be calculated using force movement analysis. Let m, $\boldsymbol{a}$, and $\boldsymbol{v}$ be the UAV's mass, acceleration vector, and velocity vector, respectively. The power associated with such movements is denoted by P_r and calculated as

$$P_r = \mathsf{m}\boldsymbol{a}^{\mathrm{T}}\boldsymbol{v}. \tag{62}$$

On the other hand, Newton's second law imposes $\xi\boldsymbol{l} = \mathsf{m}\boldsymbol{a}$, where $\xi\boldsymbol{l}$ is the lift force vector produced by all ξ propellers causing movements. Thus, Eq. (62) can be expressed as

$$\xi\boldsymbol{l}^{\mathrm{T}}\boldsymbol{v} = \mathsf{m}\boldsymbol{a}^{\mathrm{T}}\boldsymbol{v}. \tag{63}$$

Propellers produce this lift force only when they overcome pitching moment $\boldsymbol{m}$. The pitching moment-to-lift ratio $\boldsymbol{l}^{+}\boldsymbol{m}$ for a propeller is constant and equals to $\frac{\mathsf{C_M}}{\mathsf{C_L}}$, where $\boldsymbol{l}^{+}$ is the pseudo inverse defined as $\boldsymbol{l}^{+} = \frac{\boldsymbol{l}^{\mathrm{T}}}{\|\boldsymbol{l}\|^2}$ (see [73]). Thus, Eq. (62) can be expressed as [71], [72]

$$\xi\boldsymbol{l}^{\mathrm{T}}\boldsymbol{v} = \mathsf{m}\boldsymbol{a}^{\mathrm{T}}\boldsymbol{v} \xrightarrow{\times\left(\boldsymbol{l}^{+}\boldsymbol{m}\right)^{\mathrm{T}}} \xi\boldsymbol{m}^{\mathrm{T}}\boldsymbol{v} = \mathsf{m}\left(\boldsymbol{l}^{+}\boldsymbol{m}\boldsymbol{a}\right)^{T}\boldsymbol{v}$$

$$\xi\boldsymbol{m}^{\mathrm{T}}\boldsymbol{v} = \frac{\mathsf{mC_M}}{\mathsf{C_L}}\boldsymbol{a}^{T}\boldsymbol{v}. \tag{64}$$

The geometric pitch is a characteristic of a propeller and is defined as $2\pi\boldsymbol{v} = \mathsf{J}\mathsf{r}\boldsymbol{\omega}$, where $\boldsymbol{\omega}$ is the propeller's angular velocity [73]. Hence, Eq. (64) can be expressed as:

$$\xi\boldsymbol{m}^{\mathrm{T}}\boldsymbol{v} = \frac{\mathsf{mC_M}}{\mathsf{C_L}}\boldsymbol{a}^{T}\boldsymbol{v} \xrightarrow{\boldsymbol{v}=\frac{\mathsf{J}\mathsf{r}\boldsymbol{\omega}}{2\pi}} \frac{\xi\mathsf{J}}{2\pi}\boldsymbol{m}^{\mathrm{T}}\mathsf{r}\boldsymbol{\omega} = \frac{\mathsf{mC_M}}{\mathsf{C_L}}\boldsymbol{a}^{T}\boldsymbol{v}$$

$$\underbrace{\xi\boldsymbol{m}^{\mathrm{T}}\mathsf{r}\boldsymbol{\omega}}_{\alpha} = \frac{2\pi\mathsf{mC_M}}{\mathsf{JC_L}}\boldsymbol{a}^{T}\boldsymbol{v} \tag{65}$$

In Eq. (65), α represents the minimum kinetic power consumption P_a required to overcome pitching moment $\boldsymbol{m}$ and enable the UAV to accelerate $\boldsymbol{a}$ and move at velocity $\boldsymbol{v}$. Thus, due to the pitching moment, the UAV consumes P_a to supply the necessary power for movement with the power of $P_r = \mathsf{m}\boldsymbol{a}^{T}\boldsymbol{v}$, as follows:

$$P_a = \frac{2\pi\mathsf{mC_M}}{\mathsf{JC_L}}\boldsymbol{a}^{T}\boldsymbol{v} = \frac{2\pi\mathsf{C_M}}{\mathsf{JC_L}}P_r. \tag{66}$$

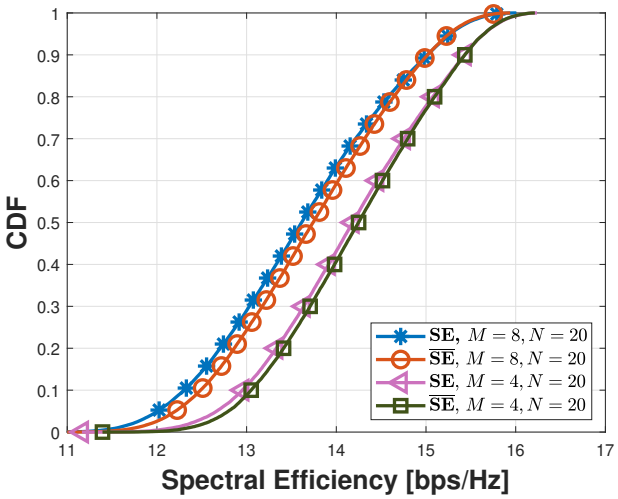


Fig. 11: $\text{SE}_m^{(n)}$ and $\overline{\text{SE}}_m^{(n)}$ for different M, N.

Appendix C

We first drop the time index and define the matrix

$$\boldsymbol{\Omega}_m = \left(\sum_{j\neq m}\hat{\boldsymbol{g}}_m\hat{\boldsymbol{g}}_m^{*}p_m + \boldsymbol{\Sigma}\right)^{-1}. \tag{67}$$

Then, defining $\boldsymbol{\Omega}'_m = N\boldsymbol{\Omega}_m$, (28) can be written as

$$\begin{aligned}\text{SINR}_m &= \hat{\boldsymbol{g}}_m^{*}\boldsymbol{\Omega}_m\hat{\boldsymbol{g}}_m\, p_m \\ &= \frac{p_m}{N}\,\text{tr}[\hat{\boldsymbol{g}}_m\hat{\boldsymbol{g}}_m^{*}\boldsymbol{\Omega}'_m]\,.\end{aligned} \tag{68}$$

For N,$M \to \infty$, using [74, Lemma 4] and [74, Theorem 1],

$$\frac{p_m}{N}\,\text{tr}[\hat{\boldsymbol{g}}_m\hat{\boldsymbol{g}}_m^{*}\boldsymbol{\Omega}'_m] - \frac{p_m}{N}\text{tr}[\boldsymbol{\Phi}_m\mathbf{T}_m] \xrightarrow{\text{a.s.}} 0. \tag{69}$$

The role of $\left(\boldsymbol{H}\boldsymbol{H}^{*} + \boldsymbol{S} + z\boldsymbol{I}_M\right)^{-1}$ in [74, Theorem 1] is played by $\boldsymbol{\Omega}'_m$. One can map the terms in [74] and our problem as follows: (*i*) $\boldsymbol{D} = \boldsymbol{\Phi}_m\,\mathrm{p}_m$, (*ii*) $\boldsymbol{R}_j = \boldsymbol{\Phi}_m\,p_m$, and (*iii*) $\boldsymbol{S} + z\boldsymbol{I}_N = \frac{1}{N}\boldsymbol{\Sigma}$. Then, matrix $\mathbf{T}_m$ follows the structure of $\boldsymbol{T}$ in [74, Theorem 1]:

$$\mathbf{T}_m = \left(\frac{1}{N}\sum_{j\neq m}^{M}\frac{\boldsymbol{\Phi}_j p_j}{1+e_j} + \frac{1}{N}\boldsymbol{\Sigma}\right)^{-1}. \tag{70}$$

Coefficients e_j can be calculated as $e_j = \lim_{t\to\infty} e_j^{(t)}$ where

$$e_j^{(t+1)} = p_j\,\text{tr}\left[\boldsymbol{\Phi}_j\left(\sum_{i\neq j}^{K}\frac{\boldsymbol{\Phi}_i p_i}{1+e_i^{(t)}} + \boldsymbol{\Sigma}\right)^{-1}\right]. \tag{71}$$

The fixed-point algorithm can be used to compute $e_j^{(n)}$. Consequently, Thm. 1 is obtained, which is validated in Fig. 11. Precisely, such figure includes the cumulative distribution functions of the ergodic and asymptotic spectral efficiencies for different M, N. Clearly, given the tightness between $\text{SE}_m^{(n)}$ and $\overline{\text{SE}}_m^{(n)}$, it can be claimed that the derived result is indeed accurate even for finite values of M and N.

REFERENCES

[1] J. Cohen, "The wildland-urban interface fire problem," *Fremontia*, vol. 38, no. 2, p. 3, 2010.

[2] G. Perry, "Current approaches to modelling the spread of wildland fire: a review," *Progress in Physical Geography: Earth and Environment*, vol. 22, no. 2, p. 222–245, 1998.

[3] M. A. Finney, *FARSITE: Fire Area Simulator-model development and evaluation*. U.S. Department of Agriculture, Forest Service, Rocky Mountain Research Station, 1998.

[4] H. Jafarkhani, "Taking to the air to help on the ground: How UAVs can help fight wildfires," in *IEEE ComSoc Technology News*, Oct. 2022.

[5] J. T. Abatzoglou and A. P. Williams, "Impact of anthropogenic climate change on wildfire across western US forests," *Proc. of the National Academy of Sciences*, vol. 113, no. 42, pp. 11770–11775, 2016.

[6] E. Chuvieco *et al.*, "Development of a framework for fire risk assessment using remote sensing and geographic information system technologies," *Ecological modelling*, vol. 221, no. 1, pp. 46–58, 2010.

[7] M. P. Thompson and D. E. Calkin, "Uncertainty and risk in wildland fire management: A review," *Journal of Environmental Management*, vol. 92, no. 8, pp. 1895–1909, 2011.

[8] E. Seraj and M. Gombolay, "Coordinated control of UAVs for human-centered active sensing of wildfires," in *2020 American Control Conf. (ACC)*, pp. 1845–1852, 2020.

[9] K. A. Ghamry *et al.*, "Cooperative control of multiple UAVs for forest fire monitoring and detection," in *IEEE Int'l Conf. on Mechatronic and Embedded Systems and Applications*, 2016.

[10] K. A. Ghamry and Y. Zhang, "Fault-tolerant cooperative control of multiple UAVs for forest fire detection and tracking mission," in *Conf. on Control and Fault-Tolerant Systems (SysTol)*, pp. 133–138, 2016.

[11] E. Seraj, A. Silva, and M. Gombolay, "Multi-UAV planning for cooperative wildfire coverage and tracking with quality-of-service guarantees," *Autonomous Agents and Multi-Agent Systems*, vol. 36, no. 2, p. 39, 2022.

[12] C. Yuan, Z. Liu, and Y. Zhang, "Fire detection using infrared images for UAV-based forest fire surveillance," in *2017 Int'l Conf. on Unmanned Aircraft Systems (ICUAS)*, pp. 567–572, IEEE, 2017.

[13] C. Yuan *et al.*, "UAV-based forest fire detection and tracking using image processing techniques," in *2015 Int'l Conf. on Unmanned Aircraft Systems (ICUAS)*, pp. 639–643, IEEE, 2015.

[14] B. L. Stevens *et al.*, *Aircraft control and simulation: dynamics, controls design, and autonomous systems*. John Wiley & Sons, 2015.

[15] S. Bouabdallah, "Design and control of quadrotors with application to autonomous flying," tech. rep., EPFL, 2007.

[16] R. W. Beard and T. W. McLain, *Small unmanned aircraft: Theory and practice*. Princeton University Press, 2012.

[17] Y. Park, W. Kim, and H. Moon, "Time-continuous real-time trajectory generation for safe autonomous flight of a quadrotor in unknown environment," *Applied Sciences*, vol. 11, no. 7, p. 3238, 2021.

[18] M. B. Milam *et al.*, "A new computational approach to real-time trajectory generation for constrained mechanical systems," in *Proc. of the 39th IEEE Conf. on Decision and Control*, pp. 845–851, 2000.

[19] C. Richter, A. Bry, and N. Roy, "Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments," in *16th Int'l Symposium ISRR*, pp. 649–666, Springer, 2016.

[20] J. L. Sanchez-Lopez *et al.*, "A real-time 3D path planning solution for collision-free navigation of multirotor aerial robots in dynamic environments," *Journal of Intelligent & Robotic Systems*, vol. 93, pp. 33–53, 2019.

[21] H. Oleynikova *et al.*, "Continuous-time trajectory optimization for online UAV replanning," in *Proceedings of the IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, pp. 5332–5339, 2016.

[22] D. Hong *et al.*, "Least-energy path planning with building accurate power consumption model of rotary unmanned aerial vehicle," *IEEE Trans. on Vehicular Tech.*, vol. 69, no. 12, pp. 14803–14817, 2020.

[23] H. V. Abeywickrama *et al.*, "Empirical power consumption model for UAVs," in *2018 IEEE 88th Vehicular Technology Conf.*, 2018.

[24] J. Zhang *et al.*, "Energy consumption models for delivery drones: A comparison and assessment," *Transportation Research Part D: Transport and Environment*, vol. 90, p. 102668, 2021.

[25] C. Diaz-Vilor, A. Lozano, and H. Jafarkhani, "Cell-free UAV networks: Asymptotic analysis and deployment optimization," *IEEE Trans. on Wireless Commun.*, vol. 22, no. 5, pp. 3055–3070, 2023.

[26] C. Diaz-Vilor, A. Lozano, and H. Jafarkhani, "Cell-free UAV networks with wireless fronthaul: Analysis and optimization," *IEEE Trans. on Wireless Commun.*, vol. 23, no. 3, pp. 2054–2069, 2024.

[27] C. Diaz-Vilor *et al.*, "Sensing and communication in UAV cellular networks: Design and optimization," *IEEE Trans. on Wireless Commun.*, vol. 23, no. 6, pp. 5456–5472, 2024.

[28] J. Guo, P. Walk, and H. Jafarkhani, "Optimal deployments of UAVs with directional antennas for a power-efficient coverage," *IEEE Trans. Commun.*, vol. 68, no. 8, pp. 5159–5174, 2020.

[29] E. Koyuncu *et al.*, "Deployment and trajectory optimization of UAVs: A quantization theory approach," *IEEE Trans. on Wireless Commun.*, vol. 17, no. 12, pp. 8531–8546, 2018.

[30] J. Guo and H. Jafarkhani, "Movement-efficient sensor deployment in wireless sensor networks with limited communication range," *IEEE Trans. on Wireless Commun.*, vol. 18, pp. 3469–3484, July 2019.

[31] J. Guo *et al.*, "A source coding perspective on node deployment in two-tier networks," *IEEE Trans. Commun.*, vol. 66, no. 7, pp. 3035–3049, 2018.

[32] F. Cheng *et al.*, "UAV trajectory optimization for data offloading at the edge of multiple cells," *IEEE Trans. Vehicular Technology*, vol. 67, no. 7, pp. 6732–6736, 2018.

[33] S. Karimi-Bidhendi, J. Guo, and H. Jafarkhani, "Energy-efficient node deployment in heterogeneous two-tier wireless sensor networks with limited communication range," *IEEE Trans. on Wireless Commun.*, vol. 20, no. 1, pp. 40–55, 2021.

[34] Y. Zeng and R. Zhang, "Energy-efficient UAV communication with trajectory optimization," *IEEE Trans. Wireless Commun.*, vol. 16, no. 6, pp. 3747–3760, 2017.

[35] Y. Zeng, J. Xu, and R. Zhang, "Energy minimization for wireless communication with rotary-wing UAV," *IEEE Trans. Wireless Commun.*, vol. 18, no. 4, pp. 2329–2345, 2019.

[36] S. Karimi-Bidhendi *et al.*, "Energy-efficient deployment in static and mobile heterogeneous multi-hop wireless sensor networks," *IEEE Trans. on Wireless Commun.*, vol. 21, no. 7, pp. 4973–4988, 2022.

[37] J. Guo and H. Jafarkhani, "Sensor deployment with limited communication range in homogeneous and heterogeneous wireless sensor networks," *IEEE Trans. on Wireless Commun.*, vol. 15, no. 10, pp. 6771–6784, 2016.

[38] H. Yang *et al.*, "Privacy-preserving federated learning for UAV-enabled networks: Learning-based joint scheduling and resource management," *IEEE J. on Sel. Areas in Commun.*, vol. 39, no. 10, pp. 3144–3159, 2021.

[39] Y. Yu *et al.*, "Multi-objective optimization for UAV-assisted wireless powered IoT networks based on extended DDPG algorithm," *IEEE Trans. on Commun.*, vol. 69, no. 9, pp. 6361–6374, 2021.

[40] R. Ding, F. Gao, and X. S. Shen, "3D UAV trajectory design and frequency band allocation for energy-efficient and fair communication: A deep reinforcement learning approach," *IEEE Trans. on Wireless Commun.*, vol. 19, no. 12, pp. 7796–7809, 2020.

[41] Z. Xia *et al.*, "Multi-agent reinforcement learning aided intelligent UAV swarm for target tracking," *IEEE Trans. on Vehicular Tech.*, vol. 71, no. 1, pp. 931–945, 2022.

[42] R. Ding *et al.*, "Trajectory design and access control for air–ground coordinated communications system with multiagent deep reinforcement learning," *IEEE IoT Journal*, vol. 9, no. 8, pp. 5785–5798, 2022.

[43] M. Sun, X. Xu, X. Qin, and P. Zhang, "AoI-energy-aware UAV-assisted data collection for IoT networks: A deep reinforcement learning method," *IEEE IoT Journal*, vol. 8, no. 24, pp. 17275–17289, 2021.

[44] D. Hong *et al.*, "Energy-efficient online path planning of multiple drones using reinforcement learning," *IEEE Trans. on Vehicular Tech.*, vol. 70, no. 10, pp. 9725–9740, 2021.

[45] Y. Wang *et al.*, "Trajectory design for UAV-based internet of things data collection: A deep reinforcement learning approach," *IEEE IoT Journal*, vol. 9, no. 5, pp. 3899–3912, 2022.

[46] Y. Li and A. H. Aghvami, "Radio resource management for cellular-connected UAV: A learning approach," *IEEE Trans. on Commun.*, vol. 71, no. 5, pp. 2784–2800, 2023.

[47] S. Kaul *et al.*, "Real-time status: How often should one update?," in *2012 Proc. IEEE INFOCOM*, pp. 2731–2735, 2012.

[48] C. Diaz-Vilor, A. Lozano, and H. Jafarkhani, "A reinforcement learning approach for wildfire tracking with UAV swarms," *arXiv:2407.05473*, 2024.

[49] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*, vol. 2. MIT press, 2018.

[50] C. Guestrin, D. Koller, and R. Parr, "Multiagent planning with factored mdps," in *Advances in Neural Information Processing Systems* (T. Dietterich, S. Becker, and Z. Ghahramani, eds.), vol. 14, MIT Press, 2001.

[51] T. Degris *et al.*, "Learning the structure of factored markov decision processes in reinforcement learning problems," in *Proc. of the 23rd Int'l Conf. on Machine Learning*, p. 257–264, 2006.

[52] V. Mnih *et al.*, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[53] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," *Advances in Neural Information Proc. Systems*, vol. 12, 1999.

[54] V. Konda and J. Tsitsiklis, "Actor-critic algorithms," *Advances in Neural Information Proc. Systems*, vol. 12, 1999.

[55] V. Mnih *et al.*, "Asynchronous Methods for Deep Reinforcement Learning," in *Int'l Conf. on Machine Learning*, pp. 1928–1937, PMLR, 2016.

[56] T. P. Lillicrap *et al.*, "Continuous Control with Deep Reinforcement Learning," *arXiv preprint arXiv:1509.02971*, 2015.

[57] S. Fujimoto *et al.*, "Addressing function approximation error in actor-critic methods," in *Int'l Conf. on ML*, pp. 1587–1596, PMLR, 2018.

[58] C. Diaz-Vilor and H. Jafarkhani, "Optimal 3D-UAV trajectory and resource allocation of DL UAV-GE links with directional antennas," *2020 IEEE Global Commun. Conf.*, pp. 1–6, 2020.

[59] C. Zhan *et al.*, "Energy-efficient data collection in UAV enabled wireless sensor network," *IEEE Wireless Communications Letters*, vol. 7, no. 3, pp. 328–331, 2018.

[60] S. Gao, H. Zhang, and S. K. Das, "Efficient data collection in wireless sensor networks with path-constrained mobile sinks," *IEEE Trans. Mobile Computing*, vol. 10, no. 4, pp. 592–608, 2011.

[61] N. Van Cuong, Y.-W. P. Hong, and J.-P. Sheu, "UAV trajectory optimization for joint relay communication and image surveillance," *IEEE Trans. on Wireless Commun.*, vol. 21, pp. 10177–10192, Dec. 2022.

[62] U. C. Çabuk *et al.*, "A holistic energy model for drones," in *2020 28th Sig. Proc. and Commun. Applications Conf.*, 2020.

[63] A. Thibbotuwawa *et al.*, "Energy consumption in unmanned aerial vehicles: A review of energy consumption models and their relation to the UAV routing," in *Proc. of 39th Int'l Conf. on Information Systems Architecture and Technology*, pp. 173–184, Springer, 2019.

[64] L. H. Manjarrez *et al.*, "Estimation of energy consumption and flight time margin for a UAV mission based on fuzzy systems," *Technologies*, vol. 11, no. 1, p. 12, 2023.

[65] M. Jacewicz *et al.*, "Quadrotor model for energy consumption analysis," *Energies*, vol. 15, no. 19, p. 7136, 2022.

[66] S. Shimamoto and Iskandar, "Channel characterization and performance evaluation of mobile communication employing stratospheric platforms," *IEICE Trans. Commun.*, vol. E89-B, no. 3, pp. 937–944, 2006.

[67] H. B. Mann and A. Wald, "On Stochastic Limit and Order Relationships," *Annals of Mathematical Statistics*, vol. 14, pp. 217–226, 1943.

[68] A. Y. Ng *et al.*, "Policy invariance under reward transformations: Theory and application to reward shaping," in *Proc. of the Sixteenth Int'l Conf. on Machine Learning*, p. 278–287, 1999.

[69] M. Schwager, B. J. Julian, M. Angermann, and D. Rus, "Eyes in the sky: Decentralized control for the deployment of robotic camera networks," *Proceedings of the IEEE*, vol. 99, no. 9, pp. 1541–1561, 2011.

[70] The 3rd Generation Partnership Project (3GPP), "Enhanced LTE support for aerial vehicles," Tech. Rep. 36.777, 3GPP, 2017.

[71] M. Eshelby, *Aircraft performance: Theory and practice*. American Institute of Aeronautics and Astronautics, Inc., 2000.

[72] M. Asselin, *An introduction to aircraft performance*. AIAA, 1997.

[73] P. G. Hill and C. R. Peterson, "Mechanics and thermodynamics of propulsion," *Reading*, 1992.

[74] S. Wagner *et al.*, "Large system analysis of linear precoding in correlated MISO broadcast channels under limited feedback," *IEEE Trans. on Inf. Th.*, vol. 58, no. 7, pp. 4509–4537, 2012.