Energy-Aware Stochastic UAV-Assisted Surveillance

Seyyedali Hosseinalipour, Student Member, IEEE, Ali Rahmati, Student Member, IEEE, Do Young Eun, Senior Member, IEEE, and Huaiyu Dai, Fellow, IEEE

Abstract—With the ease of deployment, capabilities of evading the jammers and obscuring their existence, unmanned aerial vehicles (UAVs) are one of the most suitable candidates to perform surveillance. There exists a body of literature in which the inspectors follow a deterministic trajectory to conduct surveillance, which results in a predictable environment for malicious entities. Thus, introducing randomness to the surveillance is of particular interest. In this work, we propose a novel framework for stochastic UAV-assisted surveillance that i) inherently considers the battery constraints of the UAVs, ii) proposes random moving patterns modeled via random walks, and iii) adds another degree of randomness to the system via considering probabilistic inspections. We formulate the problem of interest, i.e., obtaining the energy-efficient random walk and inspection policies of the UAVs subject to probabilistic constraints on inspection criteria of the sites and battery consumption of the UAVs, which turns out to be signomial programming that is highly non-convex. To solve it, we propose a centralized and a distributed algorithm along with their performance guarantee. This work contributes to both UAV-assisted surveillance and classic random walk literature by designing random walks with random inspection policies on weighted graphs with energy limited random walkers.

Index Terms—Unmanned aerial vehicles (UAVs), drones, surveillance, random walks, energy-aware design, Markov chains.

I. INTRODUCTION

RECENTLY, unmanned aerial vehicles (UAVs) have attracted lots of attention due to their low cost and flexibility of deployment. With the recent advances, UAVs are assumed as a promising alternative for ground robots to conduct surveillance in various applications [1]. The UAV-assisted surveillance includes collecting information, typically images or videos, about specific targets. In recent literature, the UAV is utilized as a flying camera over a given area following a pre-defined deterministic trajectory, which is already optimized with respect to the given network constraints. In particular, such systems consist of several checkpoints that UAVs are required to visit in their routes to the target location [2]. [3]. However, having deterministic trajectories for the UAVs leads to multiple security- and privacy-related concerns. In particular, the malicious entities can predict the exact locations of UAVs and also the visiting/inspection times of their areas of interest, using which they can achieve their goals while remaining unnoticed [4]–[6]. Therefore, it is of high importance and interest to leverage stochastic movement in surveillance. Another key factor to consider is that UAVs’ limited battery capacity does not allow them to fly for an unlimited period of time [7]. Hence, the designed trajectories should be both unpredictable and energy efficient.

In this paper, we propose a new model for energy-aware stochastic UAV-assisted surveillance, leveraging random walks and stochastic inspection policies. The goal is to design an optimal stochastic kernel for random walks and the inspection policies of multiple UAVs with limited battery capacities to minimize the long-term average of their energy consumption, while addressing the constraints on i) satisfying the desired long-term inspection criteria of the sites, and ii) providing a guarantee for the UAVs to return to their base nodes for battery recharging. While ensuring that the aforementioned stochastic constraints are satisfied with high probabilities, we transform this stochastic problem into obtaining the optimal Markov chain transition matrices and the inspection policies of the UAVs. We show that the resulting problem belongs to the category of highly non-convex signomial programming, which is in general intractable. To tackle this challenge, we first propose an optimal centralized algorithm that approximates the problem as a series of geometric programming problems. We demonstrate that the centralized algorithm suffers from the curse of dimensionality upon utilizing a large number of UAVs on a map with a large number of sites. We subsequently develop a consensus-based distributed algorithm combining the dual decomposition method, the average consensus algorithm, and the gradient consensus technique.

Our solution has the following characteristics: i) it explicitly considers the limited battery capacity of the UAVs and provides a reliable stochastic surveillance, in which the UAVs can return to their bases before their batteries run out with high probability; ii) it gives rise to stochastic trajectories and inspection policies, which will render exact predictions by malicious entities less likely; iii) it is energy-efficient, minimizing the long-term average energy consumption of the UAVs; and iv) it leads to soft partitioning of the map, different from the current map partitioning approaches (e.g., [8], [9]) with disjoint islands. Instead, it forms different stochastic movement patterns for the UAVs that, in general, have partial overlaps.

Our contributions in this work can be summarized as follows:

• We propose a novel framework for stochastic UAV-assisted surveillance that considers random-walk movement patterns and probabilistic inspection patterns. Our framework explicitly considers the limited battery capacities of the UAVs and allows reliable return to their base nodes.

• We formulate the problem of joint random walk and inspection policy design (in a non-Euclidean space, i.e., a graph structure) under probabilistic constraints as an optimization problem, which turns out to be a highly non-convex signomial programming problem.

• We tackle the problem by proposing a centralized algorithm based on iterative geometric programming approximation, which exploits both the method of condensation using monomial approximations and penalty functions. We theoretically investigate the optimality of our algorithm.
Given the drawbacks of the centralized algorithm, in particular the curse of dimensionality, we propose a distributed consensus-based algorithm. The cornerstone of our algorithm is a combination of the dual decomposition method, the average consensus algorithm, and the gradient consensus technique. We also investigate the convergence of our distributed algorithm and demonstrate its optimality.

It is worth noting that our proposed framework is general enough to be deployed for other applications concerned with designing energy-aware random walks, where UAV-assisted surveillance is only one use case. More precisely, we are among the first to investigate the design of energy-efficient random walks and inspection policies for multiple random walkers on a graph structure to satisfy the desired inspection criteria of the nodes, where each random walker has limited regenerative energy that gets renewed upon visiting its home base. The energy can also be interpreted as life-time or cost/budget in other applications such as data collection.

The rest of this paper is structured as follows: the system model is presented in Section II. Section III contains the Markov model analysis of the stochastic movement and inspection policies and the problem formulation. The centralized and consensus-based distributed algorithms are proposed in Section IV. Simulation results are presented in Section V. Related works are reviewed in Section VI followed by Section VII that concludes the paper and provides future directions.

II. SYSTEM MODEL

A. Network Model and Stochastic Inspections

We consider designing a UAV-assisted surveillance network for a map consisting of multiple sites. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \omega)$ denote the corresponding (weighted) network graph, where $\mathcal{V} = \{v_1, \cdots, v_{|\mathcal{V}|}\}$ denotes the set of nodes (sites), $\mathcal{E}$ denotes the set of edges, and $\omega: \mathcal{V}^2 \rightarrow \mathbb{R}^{+}$ is the weight function defined for nodes $v_i$ and $v_j$ as a function of the distance between the nodes and UAV parameters, which is further discussed in Appendix A. For each node $v_i \in \mathcal{V}$, we denote by $\gamma_i > 0$ the required inspection criterion on node $v_i$ and define it as the desired expected number of UAVs inspecting the site per unit time (in the steady state) and consider $\pi = [\pi_1, \cdots, \pi_{|\mathcal{V}|}]$.

We consider a set of UAVs $\mathcal{U} = \{u_1, \cdots, u_{|\mathcal{U}|}\}$ dedicated to conduct the surveillance, and assume $|\mathcal{U}| < |\mathcal{V}|$ to avoid triviality. Each UAV is associated with a base node (home base), to which the UAV returns after each trip to deliver its collected data and recharge its battery. Let $\nu_i \in \mathcal{V}$ and $\varphi_i > 0$ denote the base node and the full capacity of the battery of UAV $u_i$, respectively. Upon arriving at a site, a UAV can perform either of the following two actions: i) inspecting the site via turning on its sensors and cameras, ii) passing the site without conducting inspection. Thus, in our model, the energy consumption of a UAV is mainly due to the following two factors: i) maintaining levitation and physical movement between the sites, and ii) turning on the mounted sensors for inspection and data collection. For node $v_j$ and UAV $u_i$, let $I_{ij}$ denote a Bernoulli random variable, where $I_{ij} = 1$ with probability $\psi_{ij} \in (0, 1]$ indicates the inspection; and $I_{ij} = 0$ with probability $1 - \psi_{ij}$ indicates passing the node without inspection, for which $E[I_{ij}] = \psi_{ij}$. Let $\omega_i > 0$ denote the energy consumption of data collection, i.e., utilizing the camera and sensing devices, of UAV $u_i$ at node $v_j$, which can vary from one UAV to another due to heterogeneous attributes of the UAVs. We refer to Fig. I for illustrations.

B. Stochastic Movement of the UAVs

Let $X_i(t) \in \mathcal{V}$, $t = 1, 2, \cdots$ be the position of UAV $u_i$ at time $t$. In our framework, $X_i(t)$ is taken to be a Markov chain with its transition matrix $P_i = [p_{k,j}^{(i)}]_{1 \leq k,j \leq |\mathcal{V}|}$, where $p_{k,j}^{(i)} = \Pr(X_i(t+1) = v_k | X_i(t) = v_j)$ is the probability of transitioning to node $v_k$ from node $v_j$. The Markov chains of our interest are ergodic chains that admit unique stationary distributions, denoted by $\gamma_i = \{\gamma_i^{(1)}, \cdots, \gamma_i^{(|\mathcal{V}|)}\}$ for UAV $u_i$ satisfying $\gamma_i P_i = \gamma_i$. We assume that random variable $I_{ij}$, $\forall j$, is independent of the corresponding UAV’s trajectory, i.e., $p(I_{ij} = 1 | a, X_i(t) = b) = p(I_{ij} = 1 | a) = p(X_i(t) = b), \forall i, j, a, b, t$. As will be seen later, our problem is to find the optimal transition matrices and inspection policies of the UAVs to minimize their long-term average expected energy consumption while satisfying the constraints on the desired inspection criteria of the sites and the battery capacities of the UAVs. Note that the resulting trajectories of the UAVs and the inspection policies are stochastic, which, considering the existence of malicious entities, are actually desired.

III. PROBABILISTIC ANALYSIS OF THE SURVEILLANCE

A. Reliable Stochastic Surveillance with Energy Constraints

A feasible surveillance framework should fulfill the following requirements: i) the desired inspection criteria of the nodes should be satisfied; ii) the UAVs should have enough energy stored in their batteries to perform the surveillance and come between various nodes, is captured in different edge weights, representing the required energy spent for this traversal (see Appendix A). It can be verified that from the energy consumption perspective, skewing the notion of time in that manner has no effect on the analysis.

| Symbol | Definition |
|--------|------------|
| $G$    | The network graph |
| $\mathcal{V}$ | The set of nodes/sites in the network |
| $\mathcal{E}$ | The set of edges of the network graph |
| $\omega$ | The weight function |
| $v_i$ | The base node of UA V $u_i \in \mathcal{U}$ |
| $\pi$ | The desired inspection criterion of the sites |
| $\varphi_i$ | The full capacity of the battery of UA V $u_i$ |
| $\mathcal{U}$ | The set of UAVs in the network |
| $\gamma_i$ | The stationary distribution of movement of UA V $u_i \in \mathcal{U}$ |
| $\psi_{ij}$ | Energy of data collection at node $v_j \in \mathcal{V}$ for UA V $u_i \in \mathcal{U}$ |
| $X_i(t)$ | The location/position of UA V $u_i \in \mathcal{U}$ at time $t$ |
| $I_{ij}$ | The Bernoulli random variable indicating the inspection of location $v_j \in \mathcal{V}$ for UA V $u_i \in \mathcal{U}$ |
| $\tau_i$ | A random variable corresponding to the return time of the UA V $u_i$ to its base node |

\[ 1 \] The symbol $|\mathcal{V}|$ denotes the cardinality of the set.

\[ 2 \] The weight $\omega(v_i, v_j)$ can also be interpreted as the cost of transition between sites $v_i$ and $v_j$.

\[ 3 \] For instance, if $v_i$ desires to get inspected by 1 UAV every 2 time instance on average $\pi_i = 0.5$. Our formulation also accommodates the scenarios where $\pi_i > 1$ for more strict security purposes (e.g., when it is desired to get inspected by more than one UAV per time instant) and it does not require to normalize $\pi_i$-s. We ignore the trivial case of $\pi_i = 0$ since in that case node $v_i$ needs no surveillance and can be excluded.

\[ 4 \] The amount of time in transversal, which depends on the actual distance between various nodes, is captured in different edge weights, representing the required energy spent for this traversal (see Appendix A). It can be verified that from the energy consumption perspective, skewing the notion of time in that manner has no effect on the analysis.
Due to the inherent randomness in the movement of the UAVs and the inspections patterns of the sites, these constraints are of probabilistic nature. We first state these constraints and then convert them into tractable mathematical expressions to be used in our optimization framework later on.

First, assuming the Markov chain associated with the random walks to be in its stationary regime, satisfying the inspection criteria of the nodes can be expressed via upper bounding the probability of violation of the inspection criterion of each node:

\[
Pr\left(\sum_{t=1}^{T(i)} \gamma^{(i)}_{j} I^{(i)}_{j} \leq \pi_{j}\right) \leq \tilde{\theta}_{j}, \forall v_{j} \in V, \quad (1)
\]

where \(\tilde{\theta}_{j} \in (0, 1]\) is the tuning parameter controlling the amount of violation. Inequality (1) implies that the rate of inspection of site \(v_{j}\), i.e., the average number of UAVs inspecting it per unit time, is larger than \(\pi_{j}\) with high probability (at least \(1 - \tilde{\theta}_{j}\)). Second, to have reliable surveillance, each UAV should be able to visit its base node for battery recharge before its battery depletion. For each UAV, we consider the random time span between its departure from and return to its base node as a surveillance cycle. We bound the probability of exceeding the battery capacity of each UAV for each surveillance cycle:

\[
Pr\left(\sum_{t=1}^{T(i)} w(X_{i}(t), X_{i}(t+1)) + \sum_{t=1}^{T(i)} \psi_{X_{i}(t)} I^{(i)}_{X_{i}(t)} \geq \varphi_{i}, \forall u_{i} \in U, \right. \quad (2)
\]

where the first and the second terms inside the probability denote the total energy consumed for the movement and for the sensing per surveillance cycle, respectively. In (2), \(\tilde{\theta}_{i} \in (0, 1]\) is the tuning parameter controlling the tolerable amount of violation, and \(T(i)\) is a random variable corresponding to the return time of UAV \(u_{i}\) to its base node, \(T(i) = \min\{n \geq 1 : X_{i}(n) = v_{b}, X_{i}(0) = v_{b}\}\). Inequality (2) implies that the energy consumption of each UAV \(u_{i}\) during a surveillance cycle is less than \(\varphi_{i}\) with high probability (at least \(1 - \tilde{\theta}_{i}\)).

According to the strong Markov property, successive returns to a given site forms a renewal process. The energy associated with the movement of the UAV, the energy associated with using the sensing devices, or any other possible action that UAV may take during a surveillance cycle, e.g., sending and receiving data from some base stations, as the reward during the surveillance cycle. This draws a connection between the scenario considered in this paper and the framework of the reward process used to derive a tractable expression for (2).

**Theorem 1.** The sufficient conditions to satisfy the probabilistic constraints given by (1) and (2) can be expressed based on the stationary distribution of the Markov chains, the transition matrices, and the inspection probabilities of the UAVs. In particular, (1) can be transformed to:

\[
\sum_{i=1}^{\left|U\right|} \gamma^{(i)}_{j} I^{(i)}_{j} \geq \tilde{\theta}_{j} \pi_{j} + \left(1 - \tilde{\theta}_{j}\right) \sum_{i=1}^{\left|U\right|} \gamma^{(i)}_{j}, \forall v_{j} \in V, \quad (3)
\]

and (2) can be expressed as:

\[
\sum_{v_{j} \in V} \psi_{j} (1 + \tilde{\theta}_{j}) w(v_{j}, v_{k}) + \sum_{v_{j} \in V} \gamma^{(i)}_{j} I^{(i)}_{j} \leq \varphi_{i} \tilde{\theta}_{i} \quad (4)
\]

**Proof.** The proof is provided in Appendix B. 

Considering (3), by setting \(\varphi^{(i)}_{j} = 1, \forall i, j\), and taking the summation with respect to \(w.r.t.\) \(j\) from both hand sides of the inequality, the necessary condition on the number of UAVs to satisfy (3) is given by \(\left|U\right| \geq \left|\sum_{v_{j} \in V} \pi_{j}\right|\). Considering UAV \(u_{i}\), the long-term average movement energy of the UAV can be written based on its stationary distribution of the visits of the nodes and its transition matrix as:

\[
\lim_{T \to \infty} \frac{1}{T} E\left[\sum_{t=1}^{T} w(X_{i}(t), X_{i}(t+1))\right] = \sum_{v_{j} \in V, v_{b} \in V} \sum_{v_{k} \in V} \gamma^{(i)}_{j} I^{(i)}_{j} \psi_{j} (1 + \tilde{\theta}_{j}) w(v_{j}, v_{k}).
\]

In a similar manner, the UAV’s long-term average consumed energy for inspection of the nodes can be derived as:

\[
\lim_{T \to \infty} \frac{1}{T} E\left[\sum_{t=1}^{T} \psi_{X_{i}(t)} I^{(i)}_{X_{i}(t)}\right] = \sum_{v_{j} \in V} \gamma^{(i)}_{j} I^{(i)}_{j} \psi_{j} (1 + \tilde{\theta}_{j}).
\]

Thus, the long-term average consumed energy during the surveillance, is given by:

\[
\lim_{T \to \infty} \frac{1}{T} E\left[\sum_{t=1}^{T} w(X_{i}(t), X_{i}(t+1)) + \psi_{X_{i}(t)} I_{X_{i}(t)}\right] = \sum_{u_{i} \in U} \sum_{v_{j} \in V} \sum_{v_{k} \in V} \gamma^{(i)}_{j} I^{(i)}_{j} \psi_{j} (1 + \tilde{\theta}_{j}).
\]

Due to the topological structure of the problem and heterogeneous base nodes and battery capacities of the UAVs, the

\[\sum_{i=1}^{\left|U\right|} \gamma^{(i)}_{j} I^{(i)}_{j} \geq \pi_{j} \forall v_{j} \in V\]

resulting in a looser bound coinciding with (3) for \(\tilde{\theta}_{i} = 1\).
optimal transition matrices and the inspection policies of the UAVs are different. In the following, we use the above results to formulate the problem of interest.

B. Problem Formulation

Let us define the following sets: \( P = \{P_1, P_2, \cdots, P_{|\mathcal{U}|}\}, \quad \mathbf{t} = \{t_1, t_2, \cdots, t_{|\mathcal{U}|}\}, \quad \gamma = \{\gamma_1, \gamma_2, \cdots, \gamma_{|\mathcal{U}|}\}, \) where \( P_i, \gamma_i, \gamma_j \) are defined as above for UAV \( u_i, \forall u_i \in \mathcal{U}. \) The problem of interest is determining the movement and the inspection policies of the two sets \( \gamma \) and \( P \) are not independent. More precisely, given a matrix \( P_0, \) vector \( \gamma_i \) is uniquely defined, \( \forall u_i \in \mathcal{U}. \)

As a result, we perform the following change of variables:

\[
q_{jk}^{(i)} = \gamma_j^{(i)} p_{jk}^{(i)}, \quad \forall u_i \in \mathcal{U}, \quad \forall v_j, v_k \in \mathcal{V}. \quad (6)
\]

It is easy to verify the following two equations:

\[
\gamma_j^{(i)} = \sum_{v_k \in \mathcal{V}} q_{jk}^{(i)}, \quad \forall v_i \in \mathcal{U}, \quad \forall v_j \in \mathcal{V}, \quad (7)
\]

\[
\sum_{v_j \in \mathcal{V}} \sum_{v_k \in \mathcal{V}} q_{jk}^{(i)} = 1, \quad \forall u_i \in \mathcal{U}. \quad (8)
\]

Note that \( \sum_{v_j \in \mathcal{V}} p_{jk}^{(i)} = 1, \forall v_j \in \mathcal{V}, u_i \in \mathcal{U}, \) is implicitly satisfied via the above two equations. As a result, instead of finding two sets \( P \) and \( \mathbf{t} \), we focus on finding set \( q \), where \( q = \{q_1, q_2, \cdots, q_{|\mathcal{U}|}\}, \) and each \( q_i = [q_{jk}^{(i)}]_{1 \leq j, k \leq |\mathcal{V}|} \) is a matrix. For a given \( q, \) \( \gamma \) is defined as above.

Then, the transition matrices of the UAVs \( P \)'s can be obtained through \( (6) \).

Using this change of variables, with some algebraic manipulations, we formulate the energy-aware stochastic UAV-assisted surveillance as the following optimization problem:

\[
\begin{align*}
\text{arg} \min_{q, \mathbf{t}} & \quad \sum_{u_i \in \mathcal{U}} \sum_{v_j \in \mathcal{V}} \sum_{v_k \in \mathcal{V}} q_{jk}^{(i)} w(v_j, v_k) + \\
& \quad \sum_{u_i \in \mathcal{U}} \sum_{v_j \in \mathcal{V}} \sum_{v_k \in \mathcal{V}} q_{jk}^{(i)} q_{kj}^{(i)} t_j
\end{align*}
\]

s.t.\n
\[
\begin{align*}
(C1) & \quad \sum_{u_i \in \mathcal{U}} \sum_{v_j \in \mathcal{V}} q_{jk}^{(i)} t_j \geq \theta_j \pi_j \\
& \quad \quad + (1 - \theta_j) \sum_{u_i \in \mathcal{U}} \sum_{v_j \in \mathcal{V}} q_{jk}^{(i)}, \quad \forall v_j \in \mathcal{V}, \quad (15) \\
(C2) & \quad \sum_{v_j \in \mathcal{V}} q_{jk}^{(i)} w(v_j, v_k) + \sum_{v_j \in \mathcal{V}} q_{jk}^{(i)} q_{kj}^{(i)} t_j \\
& \quad \quad \leq \varphi_j \sum_{v_k \in \mathcal{V}} q_{jk}^{(i)}, \quad \forall u_i \in \mathcal{U}, \\
(C3) & \quad \sum_{v_j \in \mathcal{V}} q_{jk}^{(i)} = 1 \quad \forall v_k \in \mathcal{V}, \quad \forall u_i \in \mathcal{U}, \quad (C4) \\
(C5) & \quad 0 < q_{jk}^{(i)} \leq 1, \quad \forall v_j, v_k \in \mathcal{V}, \quad \forall u_i \in \mathcal{U}.
\end{align*}
\]

In this formulation, the objective function is the long-term average consumed energy during the surveillance given by \( \gamma_i \) is the left eigenvector of \( P_i, \forall u_i \in \mathcal{U}. \) \( \]}

The first constraint (C1) enforces the satisfaction of the desired inspection criteria presented in \( \]\), the second constraint (C2) guarantees the battery consumption presented in \( \]\), the third constraint (C3) forces the stationary visiting distribution to be the left eigenvector of the transition matrix, the forth constraint (C4) ensures that the summation of stationary visiting distribution is equal to 1 for each UAV, while the last constrain (C5) is ensuring a feasible range for the variables. A noteworthy property of (C2) is that it results in a soft partitioning of the map around the bases nodes of the UAVs, allowing the UAVs with small battery capacities to mostly inspect the sites located around their base nodes. Our choice of the term soft partitioning is due to the fact that our approach does not limit the set of nodes to do the surveillance, rather it increases the probability of surveillance for the closer nodes around the base nodes. This new perspective to the map partitioning is different from the classic map partitioning approaches in existing surveillance-related literature \( [8], [9] \). Also, our method leads to a significant decrease in the number of redundant inspections. These two facts will be further illustrated in Section IV.C.

Remark 1. Note that the probabilistic problem of interest has been transformed to jointly finding a set of transition matrices of Markov chains, each of which describing the random movement of a UAV, and a set of inspection policies subject to the constraints on the inspection criteria of the nodes (C1) and the energy consumption per surveillance cycle (C2). To the best of our knowledge, we are among the first to propose this formulation and solve it in a systematic manner.

IV. OPTIMAL RANDOM WALKS AND INSPECTION POLICIES

Solving the aforementioned optimization problem directly is non-trivial since the multiplication of the optimization variables exists in the formulation, e.g., between \( q_{jk}^{(i)} t_j \) in the objective function and in the first and the second constraints. In fact, we will show that \( (15) \) belongs to the family of signomial programming problems and is highly non-convex. To tackle this problem, we propose a tractable iterative approach, in which at each iteration we solve an approximation of the problem that has the format of geometric programming (GP).

In the following, we give a brief overview of GP.

A. Geometric Programming

A basic knowledge of monomials and posynomials, which is given below, is a prerequisite to understand the GP.

Definition 1. A monomial is defined as a function \( f : \mathbb{R}^{n+L} \rightarrow \mathbb{R} \) \( \sum_{i=1}^{n} \alpha_i y_i \) \( y = [y_1, \cdots, y_n] \), and \( \alpha_j \in \mathbb{R}, \forall y \). Further, a posynomial \( q \) is defined as a sum of monomials: \( q(y) = \sum_{m=1}^{M} \alpha_1^{(m)} y_1^{(m)} \cdots y_n^{(m)} \).

A standard GP is a non-convex optimization problem defined as minimizing a posynomial subject to posynomial inequality constraints and monomial equality constraints \( [10], [11] \):

\[
\begin{align*}
\text{min} & \quad \sum_{i=1}^{n} \alpha_i y_i \\
\text{s.t.} & \quad f_i(y) \leq 1, \quad i = 1, \cdots, I, \\
& \quad h_i(y) = 1, \quad i = 1, \cdots, L,
\end{align*}
\]

\( \mathbb{R}^{n+L}_{++} \) denote the strictly positive quadrant of \( n \)-dimensional Euclidean space.
where $f_i(y) = \sum_{j=1}^{M_i} d_{i,m} y_1^{\alpha_{i,1,m}} \cdots y_n^{\alpha_{i,n,m}}, \forall i$, and $h_l(y) = d_l y_1^{\alpha_{l,1}} \cdots y_n^{\alpha_{l,n}}, \forall l$. Since the log-sum-exp function $f(y) = \log \sum_{j=1}^{M_i} e^{y_j}$ is convex, where log denotes the natural logarithm, with the following change of variables and constants $z_i = \log(y_i)$, $b_{i,k} = \log(d_{i,k})$, $b_l = \log(d_l)$ the GP in the convex form can be obtained as:

$$ \min \sum_{i=1}^{M_0} e^{(\alpha_{i,m} z + b_{i,m})} $$

$$ \text{s.t.} \quad \sum_{m=1}^{M_0} e^{(\alpha_{i,m} z + b_{i,m})} \leq 0, \quad i = 1, \ldots, I, $$

$\alpha_i^T z + b_i = 0, \quad l = 1, \ldots, L,$

where $z = [z_1, \ldots, z_n]^T$, $\alpha_i, k = [\alpha_{i,1,k}, \alpha_{i,2,k}, \ldots, \alpha_{i,n,k}]^T$, $\forall i, k$, and $\alpha_i = [\alpha_{i,1}, \alpha_{i,2}, \ldots, \alpha_{i,n}]^T$, $\forall i, k$.

### B. Obtaining Random Walks and Inspection Policies: Centralized Approach

It can be verified that although the objective function and all the constraints in problem [15] can be expressed as monomials and posynomials w.r.t. the optimization variables, problem [15] does not obey the standard form of GP in [16]. In fact, none of the constraints (C1), (C2), (C3), and (C4) can be directly expressed as inequalities on posynomials or equalities on monomials, which is required in [16]. Thus, the problem fits into the category of signomial programming, for which direct derivation of a solution is intractable [10]. In the following, we apply two methods, namely penalty functions and monomial approximations, on problem [15], to approximate the problem as a series of GP problems. Afterward, we propose an effective algorithm to solve the problem along with its performance guarantee. We first use the method of penalty functions and auxiliary variables [12]. To this end, we consider each equality on a posynomial in the format of $g(x) = c$ as two inequality constraints: i) $g(x) \leq c$, and ii) $1 - \frac{1}{g(x)} \leq c$, where $A$ is an auxiliary variable, which will later be forced to be 1. Aiming to make problem [15] as close to a GP as possible, we perform some algebraic manipulation and rewrite it as:

9The multiplicative coefficient(s) of a posynomial/monomial should be strictly positive. For example, $-\sum_{u_i \in U} \sum_{v_{i,k} \in V} q_{i,k}^{(j)} f_j$ is not a posynomial.
Algorithm 1: Centralized random walk and inspection policy design of stochastic surveillance

Input: Convergence criterion
1. Initialize the iteration count \( m = 0 \).
2. Choose an initial feasible point \( x^{(0)} = [q^{(0)}, \ell^{(0)}] \).
3. Obtain the monomial constraints in \((10)-(14)\).
4. Replace those approximations in \((20)\).
5. Using the logarithmic approximations and taking the log from constraints, convert the GP programming in \((20)\) to a convex formulation in the form of \((17)\).
6. \( m = m + 1 \).
7. Solve the resulting convex optimization problem using an arbitrary tool to obtain the solution \( x^{[m]} \).
8. If the convergence criterion between two consecutive solutions \( x^{[m-1]} \) and \( x^{[m]} \) is not met then
9. Go to line 3 and repeat the procedure using \( x^{[m]} \).
10. else
11. Choose the obtained point as the final solution \( x^* = x^{[m]} = [q^*, \ell^*] \).
12. Replace the values of \( q^* \) in \((1)\) to obtain the optimal stationary distribution of the UAVs \( \gamma^* \).
13. Replace the values of \( \ell^* \) in \((6)\) to obtain the optimal transition matrices of the UAVs \( P^* \).

Fact 2. Assuming \( A_k^{(1)} = 1 \), and \( B_i = 1 \), \( \forall k, i \), the solution of \((18)\) coincides with the solution of \((15)\).

Problem \((18)\) still does not obey the standard GP form since the left hand sides (l.h.s) of \((C1), (C2), (C3 - 1), (C3 - 2), (C4 - 2)\) are ratios of two polynomials, instead of posynomials. Our next goal is to find favorable tight approximations for these constraints. To this end, we utilize the arithmetic-geometric mean inequality given in the following lemma.

Lemma 1 (Arithmetic-geometric mean inequality \(10, 13\)). Consider a posynomial function \( g(y) = \sum_{k=1}^{K} u_k(y) \), where \( u_k(y) \) is a monomial, \( \forall k \). The following inequality holds:

\[
\hat{g}(y) \leq \tilde{g}(y) \triangleq \prod_{k=1}^{K} \left( \frac{u_k(y)}{\alpha_k(z)} \right)^{\alpha_k(z)}, \tag{19}
\]

where \( \alpha_k(z) = u_k(z)/g(z) \), \( \forall k, \) and \( z > 0 \) is a fixed point.

We solve problem \((18)\) via an iterative approach, in which the solution at the \( m \)-th iteration \( x^{[m]} = [q^{[m]}, \ell^{[m]}] \) is obtained based on the solution of the previous iteration \( x^{[m-1]} \). We define \( h_j(x), r_i^{(k)}(x), z_k^{(j)}(x), f_k^{(j)}(x), b^{(j)}(x) \) as the denominators of \((C1), (C2), (C3 - 1), (C3 - 2), (C4 - 2)\), respectively. At each iteration, \( m \), we approximate them using the obtained solution in the previous iteration, \( x^{[m-1]} \), via arithmetic-geometric mean inequality, the result of which is given in \((10)-(14)\). It is easy to verify that \( h_j(x), r_i^{(k)}(x), z_k^{(j)}(x), f_k^{(j)}(x), b^{(j)}(x) \) given in \((10)-(14)\) are in fact the best local monomial approximations to their corresponding posynomials near fixed point \( x^{[m-1]} \) in the sense of the first-order Taylor approximation. Note that the ratio between a posynomial (e.g., the numerators of the aforementioned constraints) and a monomial (e.g., the corresponding monomial approximations of their denominators) is a posynomial. As a result, we can approximate these constraints as inequalities on posynomials at each iteration. Finally, we write the problem as \((20)\) and present the pseudo code of our proposed algorithm in Algorithm 1. The objective function and all the constraints in the following formulation obey the standard GP format. Also, the optimality of our algorithm is investigated in Proposition 1.

\[
\arg \min_{q_i \in A, B} \sum_{u_i \in U} \sum_{v_j \in V} \sum_{v_k \in V} q_{ijk} w(v_j, v_k) + \sum_{u_i \in U} \sum_{v_j \in V} q_{ijk} l(v_j) j(i) + \sum_{v_k \in V} u_k(A_k) + \sum_{u_i \in U} w_i(B_i).
\]

s.t.

\[
\frac{\hat{\theta}_j \pi_j + (1 - \hat{\theta}_j)}{\hat{h}_j(x)} \leq 1, \forall v_j \in V,
\]

\[
\frac{\sum_{v_j \in V} q_{ijk} w(v_j, v_k) + \sum_{v_k \in V} q_{ijk} l(v_j) j(i)}{\varphi_i \hat{\theta}_j \pi_j(x)} \leq 1, \forall u_i \in U,
\]

\[
\frac{\sum_{v_j \in V} h_j(x)}{\hat{B}_j} \leq 1, \forall u_i \in U,
\]

\[
\frac{\sum_{v_k \in V} h_j(x)}{b^{(j)}(x)} \leq 1, \forall u_i \in U,
\]

\[
0 < q_{ijk}, l(v_j) j(i) \leq 1, \forall v_j, v_k \in V, \forall u_i \in U.
\]

Proposition 1. Algorithm 1 generates a sequence of improved feasible solutions that converge to a point \( x^* \) satisfying the KKT conditions of the original problem formulation \((15)\).

Proof. The proof is provided in Appendix \(\square\).

C. Obtaining Random Walks and Inspection Policies: Distributed Approach

So far, our proposed method is a centralized approach, which asks for a powerful centralized processor to obtain the solution. This raises two concerns: i) the processor should have a global knowledge about all the parameters of all the UAVs, i.e., at each iteration, it should have the knowledge of the entire sets \( q = [q_1, q_2, \ldots, q_m] \) and \( \ell = [\ell_1, \ell_2, \ldots, \ell_m] \); and ii) the size of the problem is \( |U| \times |V| \times |V| \), which, for a given map, escalates quickly as the number of UAVs increases. Based on the above two considerations, we aim to develop a distributed algorithm that firstly eliminates the requirement on global knowledge and secondly is scalable w.r.t. the number of UAVs. To achieve this, we aim to break
where all the elements of each Lagrangian vector/matrix is assumed to be in $\mathbb{R}^+$. Since the problem in hand is a convex optimization problem in standard form, the duality gap is zero. As a result, the solution of problem (20) coincides with the solution of problem (24). Problem (24) can be considered as a max-min optimization problem. Throughout, we recall the inner minimization problem, i.e., deriving the dual function using (23), as the inner-problem and the outer maximization problem, i.e., deriving the Lagrangian multipliers, as the outer-problem. Based on the convexity of the original problem, which directly results in the concavity of the dual problem (24), this max-min problem can be solved iteratively by assuming a set of Lagrangian multipliers for the inner problem to obtain the corresponding solution, i.e., $\tilde{x}$, in the Lagrangian function to obtain the dual-function and solve the outer problem to find the new Lagrangian multipliers. This process can be repeated until the convergence occurs. Nonetheless, since the Lagrangian function is not a separable function w.r.t. the UAV indices, this process cannot be carried out in a distributed fashion in a straightforward manner. In the following, considering (21) we present two observations, which are of particular interest and are the cornerstones of our proposed consensus-based distributed algorithm.

**Observation 1:** Assuming known values for the Lagrangian multipliers, except for the term on the third line, all of the terms are either known or can be broken down w.r.t. the index of UAVs. However, in the term in the third line, $h_j(\tilde{x}[m-1])$ appears inside the argument of the log function, where $h_j(\tilde{x}[m-1]) = \sum_{u_i \in U} \sum_{v_k \in V} \exp \left( \left( \tilde{q}_{jk} \right)[m-1] + \tilde{r}_{jk} \right)[m-1]$, making the term inseparable w.r.t. the index $i$.

**Observation 2:** Similarly, assuming a solution $\tilde{x} = [\tilde{q}, \tilde{u}]$,
except for the third term in the second line, each term can be broken down w.r.t. the index of UAVs. Subsequently, for UAV \( u_i \), we recall \( \zeta_i, \phi_i, \rho_i, \xi_i, \delta_i, \nu_i, \chi_i, \beta_i, \epsilon_i \) as the private (local) variables and \( \lambda_i \) as a public (global) variable.

Therefore, the term in the third line of (21) makes our problem coupled and inseparable w.r.t. the UAV parameters and Lagrangian multipliers. To tackle this issue, we develop a consensus-based distributed algorithm, which consists of two steps to find the solution of the dual problem: i) solving the inner problem distributedly using distributed average consensus and the gradient descent method; ii) solving the outer problem distributedly using the consensus gradient method. To this end, we first treat each term of (21) as a (hypothetically) separate term and rewrite the Lagrangian function as follows, in which the public variable \( \lambda_j \) is replaced by a local variable \( \lambda_j^{(i)} \):

\[
L(\tilde{\lambda}, \lambda, \zeta, \phi, \rho, \xi, \delta, v, \chi, \beta, \epsilon) = \sum_{u \in U} L_i(\tilde{\lambda}, \lambda, \zeta, \phi, \rho, \xi, \delta, v, \chi, \beta, \epsilon),
\]

where function \( L_i \) is given in (22). Our distributed algorithm solves the problem through a series of GP approximations, which consists of two phases: i) for a given set of monomial approximations, obtaining the optimal solution; ii) using the obtained solution to derive the monomial approximations for the next round. The first phase itself requires solving the outer and the inner problems iteratively using gradient-based methods. In the following, obtaining each optimal solution is considered as one “iteration”, while the iterations involved in solving the outer and the inner problem are called “gradient-iteration”.

1) Solving the inner problem distributedly using distributed average consensus and the gradient descent method: Considering Observation 1, given \( h_j(\tilde{\lambda}^{(m-1)}) \), the inner-problem can be written as the sum of separable terms w.r.t. the UAVs’ indices. Consider \( h_j(\tilde{\lambda}^{(m-1)}) = \sum_{t \in I_j} h_j^{(t)}(\tilde{\lambda}^{(m-1)}) \), where \( h_j^{(t)}(\tilde{\lambda}^{(m-1)}) = \sum_{u \in V} \exp \left( \left\{ \tilde{q}_{jk}^{(m-1)} + \tilde{t}_{j}^{(m-1)} \right\} \right) \).

Note that \( h_j^{(t)}(\tilde{\lambda}^{(m-1)}) \) can be computed locally at processor \( i \). Thus, \( h_j(\tilde{\lambda}^{(m-1)}) \), the sum of those values, can be obtained distributedly using a distributed average consensus method [15–19]. Afterward, the gradient descent method can be applied locally on each term of (22). Since at iteration \( m \) the value of \( h_j(\tilde{\lambda}^{(m-1)}) \) does not change through the gradient descent updates, i.e., the gradient-iterations, it needs to be calculated once prior to solving the inner and the outer problem, and thus the consensus method does not have a significant impact on the convergence speed since the convergence is usually achieved in a few number of iterations (e.g., 55 in Section V-D).

2) Solving the outer problem distributedly using the consensus gradient method: Considering Observation 2, our approach consists of two steps: i) updating the local variables at each processor, ii) updating the global variable. Each processor first locally derives the values of the local variables by applying the gradient ascent method on (25). For example, for \( \rho_i \), at gradient-iteration \( t + 1 \), processor \( i \) performs as follows:

\[
\rho_i^{[t+1]} = \rho_i^{[t]} + c_p \left( \nabla \rho_i D(\lambda_i^{[t]}), \zeta_i^{[t]}, \phi_i^{[t]}, \rho_i^{[t]}, \xi_i^{[t]}, \delta_i^{[t]}, v_i^{[t]}, \chi_i^{[t]} , \beta_i^{[t]}, \epsilon_i^{[t]} \right),
\]

\[
\zeta_i^{[t]}, \phi_i^{[t]}, \rho_i^{[t]}, \xi_i^{[t]}, \delta_i^{[t]}, v_i^{[t]}, \chi_i^{[t]} , \beta_i^{[t]}, \epsilon_i^{[t]},
\]

where \( c_p \) is the step size. Regarding the global variable, \( \lambda_j \), \( \forall v_i, \) processor \( i \) obtains a pseudo version of it \( \lambda_j^{(i)} \) as follows:

\[
\lambda_j^{(i)} = \lambda_j^{(i)} + \epsilon \left( \nabla \lambda_j D(\lambda_j^{(i)}), \zeta_j^{[t]}, \phi_j^{[t]}, \rho_j^{[t]}, \xi_j^{[t]}, \delta_j^{[t]}, v_j^{[t]}, \chi_j^{[t]} , \beta_j^{[t]}, \epsilon_j^{[t]} \right),
\]

where the local copies of the global variable \( \lambda_j^{(i)} \) are derived by employing the consensus gradient method [20]:

\[
\lambda_j^{(i)}^{[t+1]} = \sum_{m=1}^{t} \left( W^0 \right)^{im} \lambda_j^{(m)}^{[t]},
\]

where \( W = I - cL(G_p) \), with \( L(G_p) \) the Laplacian matrix of the processors network graph \( G_p \) and \( c \in (0, 1) \), and \( \theta \in \mathbb{N} \) denotes the number of conducted consensus iterations among the adjacent processors. In this method, the adjacent processors perform \( \theta \) consensus iterations by exchanging the local copies of \( \lambda_j^{(i)} \) before updating \( \lambda_j^{(i)} \). Due to the convexity of the Lagrangian function and the concavity of the dual function, the minimax theorem [21] holds for (23) and thus the order of solving the inner and the outer problem can be interchanged.
The pseudo-code of our distributed algorithm is given in Algorithm 2. The convergence of our distributed algorithm to the KKT solution of (15) is the result of the convergence of the consensus gradient method [20] along with the convergence of the proposed GP approximation method (see Proposition 1).

V. SIMULATION RESULTS

A. Simulation Setup

We consider 200 realizations of a network graph consisting of 10 sites; the distance between the sites is chosen uniformly at random between 5km to 50km. The base node of each UAV is chosen uniformly at random among the sites, and the desired inspection criteria of the nodes \( \pi \) is a randomly generated normalized vector. We consider fixed wing UAVs moving with the average speed of 25m/s, where \( c_1 = 9.26 \times 10^{-4} \) and \( c_2 = 2.25 \times 10^3 \) in [29] [22]. If a UAV decides to inspect a site, it slows down its movement speed to 12.5m/s to conduct the inspection. The duration of inspection of each site is chosen uniformly at random between 5min to 25min to obtain the energy of data collection. Modern UAVs can be equipped with hyperspatial sensors, multi-spectral targeting systems (MTS), and light detection and ranging (LIDAR). We consider a basic application of data collection using imaging, where each UAV is equipped with a mini gyro stabilized EO/IR drone FLIR thermal imaging camera and an HD camera with power of 8W and 9W, respectively. It is assumed that the UAVs have enough battery to fly between 60km to 120km when their sensors are turned off. In the following, all the figures represent the average performance over the 200 realizations unless otherwise stated. Also, \( \theta_j = 0.9, \forall v_j \in V \) in \( \theta \), and \( \theta_i = 0.7, \forall u_i \in U \) in \( \theta \). Since there is a lack of studies devoted to investigating the stochastic surveillance for energy limited random walkers with random inspection policies, we propose the following stochastic surveillance baselines inspired by the Metropolis–Hastings (MH) Markov chain Monte-Carlo (MCMC) technique [23]-[25] combined with hard map partitioning:

1) MH-MCMC with Random Map Partitioning (MH_RMP): This baseline randomly partitions the set of network sites into disjoint subsets according to the number of UAVs, each of which contains roughly the same number of sites and only one base node. Each UAV inspects the nodes belonging to the same subset as its base. The UAVs sensors always turn on upon passing the sites. Given the normalized inspection criteria of the nodes inside each subset, the transition matrices of the Markov chains associated with the UAVs movement are obtained using the MH technique [23]-[25].

2) MH-MCMC with Random Map Partitioning and Optimized Inspection Policies (MH_RMP_OI): It follows the same procedure as MH_RMP except that it further optimizes the UAVs’ inspection policies to reduce the energy consumption.

3) MH-MCMC with Distance-based Map Partitioning (MH_DMP): This baseline is similar to MH_RMP with a different clustering rule. The nodes inside each subset are chosen to be the closest nodes (corresponding to the least movement energy) to the respective base node. The rest of the procedure is the same as MH_RMP.

4) MH-MCMC with Distance-based Map Partitioning and Optimized Inspection Policies (MH_DMP_OI): It follows the same procedure as MH_DMP except that it further optimizes the UAVs’ inspection policies.

The results presented in Sections V-B, V-C are obtained using the centralized algorithm, while the convergence of the distributed algorithm is studied in Section V-D.

B. Energy Efficiency

In Fig. 2 the top plot depicts the value of the objective function of (15), i.e., the long-term average consumed energy during the surveillance, for different numbers of UAVs w.r.t. the iteration count; the bottom plot depicts the performance comparison between our method and the baseline methods. From the top plot, it can be seen that the objective function monotonically decreases through the series of monomial approximations upon convergence, thus verifying Proposition 1. From the bottom plot, (on average) our method results in around 42% energy saving as compared to the baseline methods. In Fig. 4 we generate similar plots to Fig. 2 considering the average consumed energy per surveillance cycle. Comparing the bottom plots of Figs. 2 and 4 the performance gap between our method and the baseline methods is even more prominent, especially with fewer number of UAVs, e.g., \( |U| = 2 \) and \( |U| = 3 \), upon considering the energy consumed per surveillance cycle. This is due to the underlying map partitioning approach utilized. In our solution, UAVs that need to reach the sites located far away from the rest are usually associated with a lower surveillance cycle duration, i.e., they visit fewer sites per surveillance cycle. For example, when \( |U| = 2 \) using the baseline methods (hard map partitioning), each UAV has to inspect 5 nodes; however, this number can be different for different UAVs using our approach (see Section V-C). We break down the average consumed energy per surveillance cycle and depict the corresponding average consumed energy for movement and for inspection per surveillance in the top plot and the bottom plot of Fig. 4 respectively. A major performance gap can be seen in consumed energy for movement (top plot), which dominates the total energy consumption. Regarding the baseline methods, the distance-based partitioning of the map (MH_DMP and MH_DMP_OI) yields better performance than random map partitioning. Also, the notable effect of optimized introspection policies can be seen from Fig. 4 (bottom plot).

C. Soft Map Partitioning and Redundant Inspections

As explained earlier, our algorithm leads to soft map partitioning. This implies that the map will be probabilistically split among the UAVs, where each UAV will be present at certain partitions of the map with high probability (equivalently, most of the time). This has two main positive effects: i) it decreases the battery consumption of the UAVs; ii) it decreases the chance of redundant inspections. The latter phenomenon happens when two UAVs inspect the same site at the same time, which leads to the waste of resources. As an example, Fig. 5 depicts the final solution of our algorithm for one network realization upon having 3 UAVs. Note that i) the topology of the graph and the edge weights are omitted for better readability, and thus the respective positions of the nodes do not convey any physical information, ii) the presented result is specific and may vary from one parameter setting to another. In this figure, it is illustrated that how the satisfaction of the nodes
Average Hellinger Distance of Stationary Distributions

Number of Redundant Inspections

Long-term Average Consumed Energy

Average Consumed Energy

Average Percentage of Used Battery per SC

Fig. 2: The long-term average consumed energy during the surveillance for different numbers of UAVs w.r.t. the iteration count of our algorithm (top plot). The corresponding comparison with the baselines (bottom plot).

Fig. 3: The average consumed energy per surveillance cycle (SC) for different numbers of UAVs w.r.t. the iteration count of our algorithm (top plot). The corresponding comparison with the baselines (bottom plot).

Fig. 4: Top plot: The comparison of the average energy used for movement per surveillance cycle (SC) between our method and the baselines. Bottom plot: The comparison of the average energy used for inspection per SC between our method and the baselines.

Fig. 5: An example of soft map partitioning for a map of 10 nodes upon having 3 UAVs. The weights of the edges between the nodes are omitted for better readability.

Fig. 6: The (pairwise) average Hellinger distance between the stationary distributions of the UAVs for different numbers of UAVs.

Fig. 7: The number of redundant inspections upon letting the UAVs move for 10000 time instances for different numbers of UAVs.

Fig. 8: The average percentage of used battery per surveillance cycle (SC) for different numbers of UAVs w.r.t. the iteration count of our algorithm (top plot). The corresponding comparison with the baselines (bottom plot).

inspection criteria is achieved. As an example, the value of the inspection criterion of node 6 along with the obtained stationary distributions of the movements of the UAVs and their inspection policies are shown for node 6. It can be seen that UAV 1 will rarely leave its base node, i.e., node 2 (due to its low battery capacity and the distance of node 2 to its adjacent nodes, which are omitted for readability); the rest of the map is probabilistically partitioned among the other two UAVs to achieve the lowest average energy of surveillance. To mathematically quantify the map partitioning, we use the Hellinger distance, which for two discrete distributions \( p = (p_1, p_2, \ldots, p_n) \) and \( q = (q_1, q_2, \ldots, q_n) \) is defined as:

\[
H(p, q) = \frac{1}{\sqrt{2n}} \left( \sum_{i=1}^{n} \sqrt{p_i - q_i} \right)^2
\]

also related to the total variation distance (or statistical distance) between the two distributions.

Fig. 6 depicts the pairwise average Hellinger distance between the stationary distributions of the movement of the UAVs w.r.t. the iteration count of our centralized algorithm. Furthermore, the number of redundant inspections upon letting the UAVs move for 10000 time instances are depicted in Fig. 7. These two figures demonstrate that, as the number of iterations increases, our algorithm moves toward increasing the statistical distance between the UAVs’ movements stationary distributions, i.e., splitting the map, and decreasing the number of redundant inspections. Also, from Fig. 7 it can be noted that, initially a larger number of UAVs leads to a larger number of redundant inspections; however, when the iteration count increases, a smaller number of redundant inspections are incurred eventually due to a sharper map partitioning among the UAVs. Finally,
in the top plot of Fig. 8 we depict the average percentage of used battery of the UAVs during a surveillance cycle w.r.t. the iteration count of the algorithm; in the bottom plot the corresponding comparison with the baseline methods is depicted. From the top plot, it can be observed that after the first iteration the used battery ratios are around the upper bound set by the value of $\theta_i = 0.7, \forall u_i \in U$; however, as the iteration count increases, the usage of batteries significantly drops. This illustrates the importance of map partitioning in decreasing the energy consumption. From the bottom plot of Fig. 8 we can see that the best baseline method is MH_DMP_OI, the battery usage of which is (on average) 50% higher than that of our method. In some cases (upon having $|U| = 2$ and $|U| = 3$) the baseline methods result in more than 100% average percentage of used battery per surveillance cycle. This usually implies failure of the UAVs in the return to their bases upon using those baseline methods.

D. Convergence of the Consensus-based Algorithm

We study the convergence of our proposed distributed algorithm assuming 4 UAVs in the network. Considering the average energy of surveillance, for one network realization, Fig. 9 depicts the convergence of our distributed algorithm. Furthermore, we depict the corresponding error of convergence in Fig. 10. As can be seen, although the distributed algorithm may start from a different initial point (a different set of initial monomial approximations of the posynomials), it eventually converges to the solution of the centralized algorithm.

VI. RELATED WORKS

In a large body of existing UAV-assisted literature, the trajectory/location of the UAVs are optimized subject to network constraints, where the UAVs are considered as relaying nodes forwarding data. Considering multiple static UAVs, optimal UAV locations are derived in [26] through maximizing the data rate. We studied the optimal position planning of UAV relays considering the effect of interference in the environment [27], [28]. A UAV-assisted communication scheme is proposed in [29], where the UAV trajectory, and the transmit power of both the UAV and the mobile device are obtained to minimize the outage probability. Taking advantage of the inherent mobility feature of the UAVs, an adaptive interference avoidance position planning scheme is developed in [30]. Furthermore, trajectory design and path planning is studied with respect to power control in multi-UAV systems [31], minimizing the energy for wireless transfer (WPT)-enabled UAVs [32], and search and localization [33]. Relevant works concerning the usage of UAVs in surveillance application include multi-UAV surveillance in complex urban environments with occlusions [34], low cost vision-based indoor UAV autonomous patrolling [35], and cooperative perimeter surveillance [36]. Moreover, in [37], an energy efficient UAV surveillance scenario is investigated, where a proactive eavesdropping scheme is proposed to facilitate the eavesdropping and jamming for the legitimate UAV to maximize the amount of packets eavesdropped from the suspicious UAVs’ communication. In [38], the authors aim to maximize the average surveillance rate by optimizing the position and the jamming power of the legitimate UAV monitor.

There exists a body of literature on robotic patrolling including a realistic model of robot motion control with velocity uncertainties [39], reinforcement learning to achieve efficient cooperative behavior among the agents [40], and monitoring the locations of interest [41]. In [42], the concept of refresh time and latency of a team trajectory is introduced, where a procedure is proposed to build a road-map to represent the topological structure of the area to be patrolled. In [43], the problem of obtaining the visiting sequence of the nodes for multiple homogeneous agents is studied considering three different types of intruders. The majority of this literature focuses on studying the traveling salesman problem in different contexts. The main difference between our work and all the mentioned works lies in designing stochastic yet energy-efficient movement and inspection policies, which results in an unpredictable and secure surveillance design. Finally, a body of literature is devoted to studying different applications of random walks including community detection, spectral algorithms for approximation of the problem, and prove its optimality. We also took one step further and developed a distributed algorithm for the problem along with its performance guarantee. For future work, one can consider determining the optimal base nodes for the UAVs. Also, studying cooperative stochastic surveillance with information exchange among the UAVs seems promising.

APPENDIX A

UAV ENERGY CONSUMPTION MODELS

We present compact energy models for the fixed and rotary wing UAVs, which are easy to use in practice. We use an energy model inspired by [22], [45], [49]. Assume that at time $t = 0$, a UAV starts traveling from site $v_j$ to $v_i$, which takes $T_{ji} \in \mathbb{R}^+$ amount of time. Let $y_{ji}$ denote the corresponding physical trajectory, where $y_{ji}(t)$ is the location of the UAV at time $t \in [0, T_{ji}]$. For fixed-wing UAVs, the total propulsion energy is given by (see [22], Appendix A):

$$w(v_j, v_i) = \int_0^{T_{ji}} \left[ c_1 \|\dot{y}(t)\|^3 + c_2 \frac{\|\ddot{y}(t)\|^2}{\|\dot{y}(t)\|^2} + \frac{1}{2} m (\|\dot{y}(T_{ji})\|^2 - \|\dot{y}(0)\|^2) \right] dt$$

where $\dot{y}(t)$ and $\ddot{y}(t)$ denote the velocity and acceleration vector, respectively, $g = 9.8 \, m/s^2$, and $m$ is the mass of the UAV. Also, $c_1 = \frac{1}{2}\rho C_{d0} S$ and $c_2 = \frac{W^2}{2\rho A_{m0} S}$ are two constants, where $\rho$ is the air density in $kg/m^3$, $C_{d0}$ is the zero-lift drag coefficient of the UAV, $S$ is a reference area (e.g., the wing area), $W$ is the UAV weight in Newton, $e_0$ is the Oswald...
efficiency (typically between 0.7 and 0.85), and $A_R$ is the aspect ratio of the wing, i.e., the ratio of the wing span to its aerodynamic breadth.

For the rotary wing UAVs, in general the derivations are more complicated. Ignoring the acceleration of the UAV, the total propulsion energy is given by (see [45], Appendix):
\[
w(v_j, v_k) = \int_0^{T_i} \left[ P_0 \left( 1 + \frac{3}{2} \frac{\left( \hat{y}(t) \right)^2}{U_{tip}^2} \right) + P_1 \sqrt{1 + \frac{\left( \hat{y}(t) \right)^4}{4(\nu_0^2)} - \left( \hat{y}(t) \right)^2/2(\nu_0^2)} \right] + \frac{1}{2} \delta \rho A \left( \nu_0 \right)^3 \right] \, dt,
\]
where the first, the second and the third terms inside the integral represent the blade profile power needed to overcome the profile drag of the blades, induced energy required to overcome the induced drag of the blades, and parasitic power needed to overcome the fuselage drag, respectively. Also, $U_{tip}$ is the tip speed of the rotor balde, $\nu_0$ is the mean rotor induced velocity, $\delta$ is the profile drag coefficient, $\Omega$ is the blade angular velocity in rad/sec, $R$ is the rotor radius, $k$ is the incremental correction factor to induced power, and the rest of notations are similar to those in [29]. These physical layer expressions are used to derive the weight of the edges of the network.

**APPENDIX B**

**PROOF OF THEOREM 1**

Considering (1), to obtain a tractable expression, define the Bernoulli random variable $\tilde{f}_j^{(i)} \equiv 1 - f_j^{(i)}$. Considering the left hand side (l.h.s) of inequality (1), we get:
\[
\Pr \left( \sum_{i=1}^{[d]} \gamma_j^{(i)} f_j^{(i)} \leq \pi_j \right) = \Pr \left( \sum_{i=1}^{[d]} \gamma_j^{(i)} \tilde{f}_j^{(i)} \geq \sum_{i=1}^{[d]} \gamma_j^{(i)} \pi_j \right) \leq \frac{E \left( \sum_{i=1}^{[d]} \gamma_j^{(i)} \tilde{f}_j^{(i)} \right)}{E \left( \sum_{i=1}^{[d]} \gamma_j^{(i)} \pi_j \right)}
\]
where the inequality is the result of the Markov inequality. Note that $\sum_{i=1}^{[d]} \gamma_j^{(i)} \geq \pi_j$, $\forall \psi_j \in \mathcal{V}$, is implicitly assumed, and in fact it will be satisfied in the final solution; since otherwise the inspection criteria of the nodes cannot be satisfied even if all the UAVs turn on their sensing devices all the time. Using the above equation, the result of Theorem 1 can be obtained.

To derive a tractable expression for (2), we use the following lemma and the result of the renewal reward theorem.

**Lemma 2 (Mean return time).** Consider UAV $u_i$ with return time $T_i^{+}$. Given that the UAV starts the surveillance from its base node, i.e., $X_i(0) = v_0$, we have [50]: $E_i[T_i^{+}] = 1/\gamma_{b_i}$.

**Definition 2 (Reward process).** Consider a counting process $(N(t) : t \geq 0)$ associated with i.i.d. inter renewal times $(X_n : n \in \mathbb{N})$ having common distribution $F$. At the end of the $n$-th renewal interval, a random reward $R_n$ is earned. Let $(X_n, R_n)$ be i.i.d. with the reward $R_n$ possibly dependent on $X_n$. Then the reward process $(R(t) : t \geq 0)$ consists of accumulated reward earned by time $t$ as $R(t) = \sum_{i=1}^{N(t)} R_i$.

**Theorem 2 (Renewal reward theorem [51], [52]).** Let $N(t)$ be a counting process associated with $(X_n, R_n)$, $n \geq 1$. Assuming $r = E[R_i] < \infty$ and $\tau = E[X_1] < \infty$, we have:
\[
\lim_{t \to \infty} \frac{E[R(t)]}{t} = \frac{r}{\tau}.
\]

Let us define $E[M_i] \triangleq E \left[ \sum_{i=1}^{T_i^{+}} w(X_i(t), X_i(t+1)) \right]$ and $E[L_i] \triangleq E \left[ \sum_{i=1}^{T_i^{+}} |X_i(t)|^2 I_{X_i(t)}(X_i(t+1)) \right]$. In other words, $E[M_i]$ and $E[L_i]$ refer to the expected value of the movement energy and the expected value of the inspection energy per surveillance cycle, respectively. In the following, we derive a closed-form expression for each of them in order. For UAV $u_i$, the long-term average expected energy of movement is given by:
\[
\lim_{T \to \infty} \frac{1}{T} E \left[ \sum_{t=1}^{T} w(X_i(t), X_i(t+1)) \right] = \sum_{v_j \in \mathcal{V}, v_k \in \mathcal{V}} \gamma_j^{(i)} p_{jk}(i) w(v_j, v_k).
\]

On the other hand, we can obtain the following expression:
\[
\lim_{T \to \infty} \frac{1}{T} E \left[ \sum_{t=1}^{T} |X_i(t)|^2 I_{X_i(t)}(X_i(t+1)) \right] = \frac{E[M_i]}{E[T_i^{+}]} = \frac{E[M_i]}{1/\gamma_{b_i}}.
\]

Comparing the above two equations, we get:
\[
E[M_i] = \frac{1}{\gamma_{b_i}} \sum_{v_j \in \mathcal{V}, v_k \in \mathcal{V}} \gamma_j^{(i)} p_{jk}(i) w(v_j, v_k).
\]

Also, considering UAV $u_i$, the long-term average expected energy of nodes inspections can be expressed as:
\[
\lim_{T \to \infty} \frac{1}{T} E \left[ \sum_{i=1}^{T_i^{+}} |X_i(t)|^2 I_{X_i(t)}(X_i(t+1)) \right] = 1/\gamma_{b_i}.
\]

Using Theorem 2 and Lemma 2 we get:
\[
\lim_{T \to \infty} \frac{1}{T} E \left[ \sum_{i=1}^{T_i^{+}} |X_i(t)|^2 I_{X_i(t)}(X_i(t+1)) \right] = \frac{E[L_i]}{E[T_i^{+}]} = \frac{E[L_i]}{1/\gamma_{b_i}}
\]

Comparing the above two equations, we get:
\[
E[L_i] = \sum_{v_j \in \mathcal{V}, \psi \in \mathcal{V}} \gamma_j^{(i)} |\psi_j|^{1/\gamma_{b_i}}.
\]

Considering the l.h.s. of (2), using Markov inequality, we get:
\[
\Pr \left( \sum_{i=1}^{T_i^{+}} w(X_i(t), X_i(t+1))+ \sum_{i=1}^{T_i^{+}} |X_i(t)|^2 I_{X_i(t)}(X_i(t+1)) \right) \leq E[M_i] + E[L_i].
\]

After replacing the results of (37) and (40) in the above equation, the result of Theorem 1 can be obtained.

**APPENDIX C**

**PROOF OF PROPOSITION 1**

We first prove that algorithm [1] generates a sequence of improved feasible solutions that converge to a point $\bar{x}$ satisfying the KKT conditions of (18). Note that [20] is in fact an inner approximation of [18] [53]. Hence, it is sufficient to prove the following three characteristics for [20] [53]:

1. **Feasibility:** The solution $\bar{x}$ satisfies the feasibility constraints (18) of the optimization problem.
2. **Feasibility:** The solution $\bar{x}$ satisfies the feasibility constraints (18) of the optimization problem.
3. **Feasibility:** The solution $\bar{x}$ satisfies the feasibility constraints (18) of the optimization problem.
\[ \hat{h}_j(\mathbf{x}^{[m]}) = \prod_{u_i \in \mathcal{U}_j \forall v \in \mathcal{V}} \left( \frac{\psi_{ijk}^{(1)} \psi_{ijk}^{(2)} \psi_{ijk}^{(3)} h_j(\mathbf{x}^{[m]})}{\eta_{ijk}^{(1)} \eta_{ijk}^{(2)} \eta_{ijk}^{(3)}} \right) = \hat{h}_j(\mathbf{x}^{[m]}) \left( \prod_{u_i \in \mathcal{U}_j \forall v \in \mathcal{V}} \frac{\psi_{ijk}^{(1)} \psi_{ijk}^{(2)} \psi_{ijk}^{(3)} h_j(\mathbf{x}^{[m]})}{\eta_{ijk}^{(1)} \eta_{ijk}^{(2)} \eta_{ijk}^{(3)} h_j(\mathbf{x}^{[m]})} \right) \]

\[ \frac{\partial (g(\mathbf{x}))}{\partial x_i} \bigg|_{g(\mathbf{x}) = g(\mathbf{x}^{[m]})} = \frac{\partial (g(\mathbf{x}))}{\partial x_i} \bigg|_{g(\mathbf{x}) = g(\mathbf{x}^{[m]})} \cdot \left( \frac{\partial (\hat{g}(\mathbf{x}))}{\partial x_i} \bigg|_{\hat{g}(\mathbf{x}) = \hat{g}(\mathbf{x}^{[m]})} \right)^2 \]

\[ \frac{\partial (\hat{h}_j(\mathbf{x}))}{\partial x_i} = \left( \frac{\partial (g(\mathbf{x}))}{\partial x_i} \right)^2 \bigg|_{g(\mathbf{x}) = g(\mathbf{x}^{[m]})} \]

1) All the approximations conducted in (18) should result in tightening the constraints in (18), mathematically, for every inequality in the original problem (18) in the form of \( v\mathbf{x} \leq 1 \) and its approximated version \( \tilde{v}\mathbf{x} \leq 1 \) in (20), we should have \( v\mathbf{x} \leq \tilde{v}\mathbf{x} \). Considering C1 and C1 as an example, we get:

\[ \hat{h}_j(\mathbf{x}) \geq h_j(\mathbf{x}) \Rightarrow \]

\[ \hat{h}_j(\mathbf{x}) \geq h_j(\mathbf{x}) \]

2) The equality of the constraints in (20) should be satisfied upon convergence; mathematically, for every inequality in the original problem (18) in the form of \( v\mathbf{x} \leq 1 \) and its approximated version \( \tilde{v}\mathbf{x} \leq 1 \) in (20), we should have \( v\mathbf{x} = \tilde{v}\mathbf{x} \). As an example, we prove that this holds between C1 and C1 in (32). Note that since the numerators of the two constraints are the same, examining the equality of the denominators is sufficient. The proof for the rest of the constraints is similar and omitted for brevity.

3) The KKT conditions of (18) should be satisfied after the series of approximations converges in (20); mathematically, for every inequality in the original problem (18) in the form of \( v\mathbf{x} \leq 1 \) and its approximated version \( \tilde{v}\mathbf{x} \leq 1 \) in (20), we should have \( \nabla v\mathbf{x}^m = \nabla \tilde{v}\mathbf{x}^m \). In (33), we prove this for a general approximation of the ratio of two polynomials, where \( \tilde{v}\mathbf{x} \) is approximated by \( \hat{v}\mathbf{x} \) and \( \hat{g} \) has the format given in (19). The proof for the rest of partial derivatives, and thus the gradient, is similar.

The proof of the proposition is the direct consequence of combining the above result with Fact 1 and Fact 2.

REFERENCES

[1] J. Gu, T. Su, Q. Wang, X. Du, and M. Guizani, “Multiple moving targets surveillance based on a cooperative network for multi-UAV,” IEEE Commun. Mag., vol. 56, no. 4, pp. 82–89, 2018.

[2] J. Scherer and B. Rinner, “Persistent multi-UAV surveillance with energy and communication constraints,” in Proc. IEEE Int. Conf. Auto. Sci. Eng. (CASE), Aug 2016, pp. 1225–1230.

[3] A. Trotta, F. D. Andreagiovanni, M. Di Felice, E. Natalizio, and K. R. Chowdhury, “When UAVs ride a bus: Towards energy-efficient city-scale video surveillance,” in Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM), April 2018, pp. 1043–1051.

[4] R. W. Beard, T. W. McLain, D. B. Nelson, D. Kingston, and D. Johanson, “Decentralized cooperative aerial surveillance using fixed-wing miniature UAVs,” Proc. IEEE, vol. 94, no. 7, pp. 1306–1324, 2006.

[5] X. Yue, Y. Liu, J. Wang, H. Song, and H. Cao, “Software defined radio and wireless acoustic networking for amateur drone surveillance,” IEEE Commun. Mag., vol. 56, no. 4, pp. 90–97, 2018.

[6] G. Gu and J. Tang, “Survivable networks via UAV swarms guided by decentralized real-time evolutionary computation,” arXiv preprint arXiv:1902.07860, 2019.

[7] N. H. Motlagh, M. Bagaa, and T. Taleb, “UAV-based IoT platform: A crowd surveillance use case,” IEEE Commun. Mag., vol. 55, no. 2, pp. 128–134, 2017.

[8] L. Lin, Y. Lu, Y. Pan, and X. Chen, “Integrating graph partitioning and matching for trajectory analysis in video surveillance,” IEEE Trans. Image Process., vol. 21, no. 12, pp. 4844–4857, 2012.

[9] S. Sastry, D. Culler, M. Howard, T. Roosta, B. Zhu, J. Taneja, S. Kim, S. Schaffert, J. Hui, P. Dutta et al., “Instrumenting wireless sensor networks for real-time surveillance,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), 2006, pp. 3128–3133.

[10] M. Chiang, “Geometric programming for communication systems,” Found. Trends® Commun. Inf. Theory, vol. 2, no. 1–2, pp. 1–154, 2005.

[11] S. Boyd, S.-J. Kim, L. Vandenberghe, and A. Hassibi, “A tutorial on geometric programming,” Optim. Eng., vol. 8, no. 1, p. 67, 2007.

[12] G. Xu, “Global optimization of sigmoidal geometric programming problems,” Eur. J. Oper. Res., vol. 233, no. 3, pp. 500–510, 2014.

[13] R. J. Duffin and E. L. Peterson, “Reversed geometric programs treated by harmonic means,” Indiana Univ Math J., vol. 22, pp. 531–550, 1972.

[14] S. Boyd and L. Vandenberghe, Convex optimization. Cambridge Univ. Press, 2004.

[15] L. Xiao, S. Boyd, and S.-J. Kim, “Distributed average consensus with least-mean-square-deviation,” J. Parallel Distrib. Comput., vol. 67, no. 1, pp. 33–46, 2007.

[16] T. C. Aysal, M. Coates, and M. Rabbat, “Distributed average consensus using probabilistic quantization,” in Proc. IEEE/SP 14th Workshop Stat. Signal Process., Aug 2007, pp. 640–644.

[17] S. Patterson, B. Bamieh, and A. El Abbadi, “Distributed average consensus with stochastic communication failures,” in Proc. 46th IEEE Conf. Decis. Cont., Dec 2007, pp. 4215–4220.

[18] T. C. Aysal, B. N. Oreshkin, and M. J. Coates, “Accelerated distributed average consensus via localized node state prediction,” IEEE Trans. Signal Process., vol. 57, no. 4, pp. 1563–1576, April 2009.

[19] L. Xiao, S. Boyd, and S. Lall, “Distributed average consensus with time-varying metropolis weights,” Automatica, 2006.

[20] B. Johansson, T. Keviczky, M. Johansson, and K. H. Johansson, “Subgradient methods and consensus algorithms for solving convex optimization problems,” in Proc. 47th IEEE Conf. Decis. Control (CDC), 2008, pp. 4185–4190.

[21] K. Fan, “Minimax theorems,” Proc. National Academy Sci. United States of America, vol. 39, no. 1, p. 42, 1953.

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

[23] C. Lee, J. Kwak, and D. Y. Eun, “Towards distributed optimal movement strategy for data gathering in wireless sensor networks,” IEEE Trans. Parallel Distrib. Syst., vol. 27, no. 2, pp. 574–584, Feb 2016.
