IEEE Control Systems Letters paper presented at
2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Distributed Markov Chain-based Strategies for Multi-Agent Robotic Surveillance

Gilberto Díaz-García*, Francesco Bullo*, Jason R. Marden*

***Abstract*— Markov chains have been increasingly used to define persistent robotic surveillance schemes. Motivations for this design choice include their easy implementation, unpredictable surveillance patterns, and their well-studied mathematical background. However, applying previous results to scenarios with multiple agents can significantly increase the dimension of the problem, leading to intractable algorithms. In this work we analyze the hitting time minimization problem for multiple agents moving over a finite graph. We exploit the structure of this problem to propose a tractable algorithm to design Markov chains to cover the graph with multiple interacting agents. Using mathematical analysis, we provide guarantees for the convergence of our proposed solution. Also, through numerical simulations, we show the performance of our approach compared to the current state of art in multi-agent scenarios.**

I. INTRODUCTION

Persistence surveillance describes a task where one or multiple robotic agents are commanded to visit a set of different sites of interest. Given that the access to those locations can be restricted, the space is usually described by a finite undirected graph. Under this setting, we can describe stochastic patrolling schemes as random walks using a discrete-time Markov chains. Using Markov chains provides a mathematical framework for the design of surveillance strategies that includes unpredictability in its design, a desirable trait in scenarios with potential adversaries. Therefore, there is an increasing interest for designing algorithms to define Markov chains for multiple patroller agents.

Depending on if a model for the intruder is defined or not, Markov chain-based algorithms can be classified as adversarial-based or metric-based [1]. On the adversarial-based design, the surveillance strategies are designed such that they reduce the effect of the intruder disruption under some assumptions on its behavior. Some remarkable adversarial models consider capabilities of the intruder to move over the network [2], needed time to have a successful intrusion [3], knowledge of the patrolling strategy [4], limited observation time for the adversary [5], among other traits. Usually these scenarios can be modeled either as a stochastic game [6] or Stackelberg game [7] which uses game theoretic analysis to find optimal strategies that maximize the expected utility of the defender.

On the other hand, metric-based algorithms try to maximize the performance of the proposed surveillance strategy under particular characteristics of it. Generally, they include, but are not limited to, covering speed, visit frequency and predictability. Some relevant metrics used in the literature include time elapsed since the last visit or idleness [8], weighted mean first passage times or hitting times [9], entropy of the proposed Markov chains [10], speed of convergence or mixing rates and randomness of the motion of the patrollers [11]. For these design algorithms, the solution is obtained via an optimization program. However, as the number of patrollers increases, the complexity of the optimization problem also increases making the problem quickly intractable [12].

In this article, we focus on the tractability of the optimization problem associated with the hitting time for multiple agents. With this in mind, we list the main contributions of this document as follows. First, we formulate an alternative definition for the hitting time of a Markov chain that can be easily extendable to the multi-agent case. Second, we propose optimization problems to find the optimal Markov chains, for both the single-agent and multi-agent case, and the advantages and potential issues with each one. Third, we present a distributed algorithm that minimizes the average hitting time for multiple patrollers. By exploiting the structure of the multi-agent formulation, we are able to define an iterative optimization problem for each one of the agents that is equivalent to the centralized hitting time minimization problem. With this approach we are not only able to distributively design patrolling strategies but also reduce the complexity of doing it. Moreover, we provide mathematical analysis that guarantees the convergence of our proposed solution. Finally, we apply our solution to an archetypal surveillance problem to illustrate the capabilities of our solution. Using two different data sets, for San Francisco and Minneapolis road maps, we utilize our algorithm in scenarios with at most 700 locations and 50 patrolling agents to verify not only its performance at the surveillance task but also its ability to reduce the overall complexity of the resulting optimization problems.

II. PRELIMINARIES

In this section, we introduce some concepts and previous results about discrete-time Markov chains. A Markov chain is a sequence of random variables X_k for $k \geq 0$ defined over the state space $\mathcal{S} = \{1, \ldots, n\}$ such that the Markov property is satisfied, i.e., $\mathbb{P}[X_{k+1} = s_{k+1} | X_k = s_k, \ldots, X_0 = s_0] = \mathbb{P}[X_{k+1} = s_{k+1} | X_k = s_k]$ with $s_i \in \mathcal{S}$. Each Markov chain has an associated transition matrix $P \in \mathbb{R}^{n \times n}$ where the $(i, j)-$component describes the transition probability from state i to state j, $p_{ij} = \mathbb{P}[X_{k+1} = j | X_k = i]$.

We define I_n as the identity matrix of size n. For any square matrix $A \in \mathbb{R}^{n \times n}$, $\lambda_i(A)$ denote its ith eigenvalue and $\rho(A) = \max_i |\lambda_i(A)|$ its spectral radius. For every matrix

A with $\rho(A) < 1$ we have that $\sum_{k=0}^{\infty} A^k = (I_n - A)^{-1}$. If all the eigenvalues of A are real, we can define its maximum eigenvalue as $\lambda_{\max}(A) = \max_i \lambda_i(A) \leq \rho(A)$. Moreover, any quadratic form $x^\top Ax$ can be upper bounded by $\lambda_{\max}(A)\,||x||^2$ with $||x||$ the norm of vector x.

A Markov chain is *irreducible* if every state can be reached from any other state. Any irreducible Markov chain P has a unique stationary distribution $\pi \in \mathbb{R}^n$ such that $\pi^\top P = \pi^\top$ and $\pi(i) > 0$ for all $i \in \mathcal{S}$. In addition, a Markov chain is *reversible* if $\pi(i)p_{ij} = \pi(j)p_{ji}$ for all $(i,j) \in \mathcal{S}^2$. If we define $D = \operatorname{diag}(\pi)$ then the reversibility condition can be written as $DP = P^\top D$. Furthermore, the stationary distribution π matches the time average limiting distribution, i.e., the stationary distribution π encodes the average time spent for each state $i \in \mathcal{S}$ in the long run.

A graph is a tuple $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the edge set that describes the interaction between nodes. Let $|A|$ denote the cardinality of set $\mathcal{A}$. We say that a Markov chain is *supported on* $\mathcal{G}$ if the state space corresponds to the nodes of $\mathcal{G}$, $\mathcal{S} = \mathcal{V}$, and the only allowed transitions are those defined over the edges of $\mathcal{G}$, $(i,j) \notin \mathcal{E} \implies p_{ij} = 0$. We allow self-loops on the graphs since $p_{ii} > 0$ for some $i \in \mathcal{S}$.

For any Markov chain, the first hitting time from state i to the set $\mathcal{A} \subseteq \mathcal{S}$ is the random variable defined as, $T_{\mathcal{A}}(i) = \min\{k \geq 0 \mid X_0 = i \text{ and } X_k \in \mathcal{A}\}$. The expected value $h_{\mathcal{A}}(i) = \mathbb{E}[T_{\mathcal{A}}(i)]$ is called the mean hitting time from state i to the set $\mathcal{A}$ and it is defined as [13],

$$h_{\mathcal{A}}(i) = \begin{cases} 0 & \text{if } i \in \mathcal{A}, \\ 1 + \sum_{j \notin \mathcal{A}} p_{ij} h_{\mathcal{A}}(j) & \text{if } i \notin \mathcal{A}. \end{cases} \tag{1}$$

Let us define $\delta_{\mathcal{A}} \in \mathbb{R}^n$ as $\delta_{\mathcal{A}}(i) = 1$ when $i \notin \mathcal{A}$ and $\delta_{\mathcal{A}}(i) = 0$ otherwise; and $E_{\mathcal{A}} = \operatorname{diag}(\delta_{\mathcal{A}}) \in \mathbb{R}^{n \times n}$. Therefore, definition in Equation (1) is equivalent to,

$$h_{\mathcal{A}} = \delta_{\mathcal{A}} + E_{\mathcal{A}} P E_{\mathcal{A}} h_{\mathcal{A}} \implies h_{\mathcal{A}} = (I_n - E_{\mathcal{A}} P E_{\mathcal{A}})^{-1} \delta_{\mathcal{A}}. \tag{2}$$

Note that for any $\mathcal{A} \subseteq \mathcal{S}$ with $|\mathcal{A}| \geq 1$, we have $\rho(E_{\mathcal{A}} P E_{\mathcal{A}}) < 1$ so $I_n - E_{\mathcal{A}} P E_{\mathcal{A}}$ is positive definite. Equivalently, the definition in Equation (2) for expected hitting time is well-posed. Then, we can define the average hitting time for $\mathcal{A}$ as,

$$m_{\mathcal{A}}(P) = \pi^\top h_{\mathcal{A}} = \pi^\top (I_n - E_{\mathcal{A}} P E_{\mathcal{A}})^{-1} \delta_{\mathcal{A}}. \tag{3}$$

The average hitting time $m_{\mathcal{A}}(P)$ characterize the expected time to reach a state in $\mathcal{A}$ from any other state in the Markov chain. In robotic surveillance, this metric reflects the efficiency of a Markov chain with transition probability matrix P to cover the locations contained in $\mathcal{A}$.

III. Designing Markov Chains for Surveillance

We now present a family of optimization problems for robotic surveillance. We aim to visit in an unpredictable order a set of locations $\mathcal{A}$ as quickly as possible. First, assume that the locations to surveil are described by an undirected connected graph $\mathcal{G}$ where the edges represent the paths between any pair of locations. To cover all the locations in finite time, we constrain our design to irreducible stochastic matrices. Let us denote $\mathbb{M}_\pi$ as the set of irreducible stochastic matrices with stationary distribution π. In Equation (4), we propose an optimization problem to find a Markov chain that minimizes the average hitting time for any given supporting graph $\mathcal{G}$ and stationary distribution π.

$$\begin{array}{ll} \min\limits_{P \in \mathbb{M}_\pi} & \pi^\top (I_n - E_{\mathcal{A}} P E_{\mathcal{A}})^{-1} \delta_{\mathcal{A}} \\ \text{subject to} & p_{ij} = 0 \quad \forall\ (i,j) \notin \mathcal{E} \end{array} \tag{4}$$

For Equation (4) and later optimization problems, we define the problem size to be the number of states of the Markov chain. Thus, the problem shown in Equation (4) has size $O(n)$. However, the function $P \mapsto \pi^\top (I_n - E_{\mathcal{A}} P E_{\mathcal{A}})^{-1} \delta_{\mathcal{A}}$ is non-convex. Let $\mathbb{M}^*_\pi$ denote the set of irreducible and reversible stochastic matrices with stationary distribution π. Optimizing over $\mathbb{M}^*_\pi$ preserves convexity of the feasible set in Problem (4). Also, reversible Markov chains are highly entropic [10], a desirable property in surveillance tasks due their unpredictability. In addition, the function $m_{\mathcal{A}}(P)$ becomes convex over $\mathbb{M}^*_\pi$ turning problem in Equation (4) into a convex program.

Lemma 3.1: Convexity of Average Hitting Time. For any non-empty $\mathcal{A}$ the average hitting time $m_{\mathcal{A}}(P)$ is a convex function over $P \in \mathbb{M}^*_\pi$.

Proof: Recall that the reversibility assumption can be written as $DP = P^\top D$. Let us define the symmetric matrix $Q := D^{\frac{1}{2}} P D^{-\frac{1}{2}}$. Then, $(I_n - E_{\mathcal{A}} P E_{\mathcal{A}})^{-1} = D^{-\frac{1}{2}} (I_n - E_{\mathcal{A}} Q E_{\mathcal{A}})^{-1} D^{\frac{1}{2}}$ and $(I_n - E_{\mathcal{A}} P E_{\mathcal{A}})^{-1} \delta_{\mathcal{A}} = E_{\mathcal{A}} (I_n - E_{\mathcal{A}} P E_{\mathcal{A}})^{-1} \delta_{\mathcal{A}}$. Thus, we can reformulate the definition of average hitting time as follows,

$$\begin{aligned} m_{\mathcal{A}}(P) &= \pi^\top (I_n - E_{\mathcal{A}} P E_{\mathcal{A}})^{-1} \delta_{\mathcal{A}} = 1^\top D E_{\mathcal{A}} (I_n - E_{\mathcal{A}} P E_{\mathcal{A}})^{-1} \delta_{\mathcal{A}} \\ &= \delta_{\mathcal{A}}^\top D^{\frac{1}{2}} (I_n - E_{\mathcal{A}} Q E_{\mathcal{A}})^{-1} D^{\frac{1}{2}} \delta_{\mathcal{A}} = g\left(D^{\frac{1}{2}} \delta_{\mathcal{A}}, I_n - E_{\mathcal{A}} Q E_{\mathcal{A}}\right), \end{aligned}$$

with $g(x, Y) = x^\top Y^{-1} x$ the matrix fractional function, which is convex over $\mathbb{R}^n \times \mathbb{S}^n_{++}$ with $\mathbb{S}^n_{++}$ the set of positive definite matrices [14]. Since $\rho(E_{\mathcal{A}} Q E_{\mathcal{A}}) < 1$ then we can ensure that $m_{\mathcal{A}}(P)$ is a convex function. ■

To surveil a collection of subsets $\mathcal{C} = \{\mathcal{A}_1, \ldots, \mathcal{A}_s\}$ we can define the *worst-case average hitting time*,

$$m^{\text{wc}}_{\mathcal{C}}(P) = \max_{\mathcal{A} \in \mathcal{C}} \pi^\top (I_n - E_{\mathcal{A}} P E_{\mathcal{A}})^{-1} \delta_{\mathcal{A}}, \tag{5}$$

or the *weighted average hitting time*,

$$m^{\text{avg}}_{\mathcal{C}}(P) = \sum_{\mathcal{A} \in \mathcal{C}} \gamma_{\mathcal{A}}\, \pi^\top (I_n - E_{\mathcal{A}} P E_{\mathcal{A}})^{-1} \delta_{\mathcal{A}}, \tag{6}$$

with $\gamma_{\mathcal{A}} \geq 0$ for all $\mathcal{A} \in \mathcal{C}$. Functions in Equations (5) and (6) consider the average hitting times for all $\mathcal{A}_i \in \mathcal{C}$ and their convexity comes from the convexity of $m_{\mathcal{A}}(P)$. Consequently, without losing any generality, we focus our discussion on the average hitting time for subset $\mathcal{A}$.

IV. Successive Upper Bound Optimization Approach for Multiple Patrollers

In this section, we extend the formulation of the optimal robotic surveillance problem for multiple patrollers. Similar to the single-agent version of the problem, we want to cover a set of locations $\mathcal{A} \subseteq \mathcal{V}$ as fast as possible. However, now we have a set $\mathcal{M}$ of m different patrollers that can visit the locations in $\mathcal{V}$. We take into account that each agent can have different capabilities to move over the locations by assuming that each agent has it own supporting graph $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)$ with $\mathcal{V}_i \subseteq \mathcal{V}$ for each $i \in \mathcal{M}$. Since these capabilities are an intrinsic property of the agents, we assume that the graphs $\mathcal{G}_i$ are known a priori and they must be considered in the design of the patrolling strategy. Furthermore, the possible existence of overlapping graphs supposes a challenge in the strategy design since it requires the coordinated effort of multiple patrollers to effectively reduce the hitting time.

To tackle the multi-agent patrolling problem, a common method is to use the formulation on Section III using Markov chains with state space $\mathcal{S} = \times_{i\in\mathcal{M}} \mathcal{V}_i$. However, such Markov chains induce a coupled behavior since the transition between any pair of states in $\mathcal{S}$ is determined by the actions of more than one agent. In addition, the state space for these Markov chains $\mathcal{S}$ has size $O(n^m)$ and grows exponentially with the number of patrollers.

With this in mind, an efficient approach is to constrain our design to decoupled Markov chains for each agent. In those Markov chains, the transition probability for every agent $i \in \mathcal{M}$ depends upon only its own current state $s_i \in \mathcal{S}_i$ and it is independent of others agents states $s_j \in \mathcal{S}_j$ with $i \neq j$. Therefore, if each agent has its own supporting graph $\mathcal{G}_i$ then, we will design an irreducible and reversible Markov chain P_i defined over $\mathcal{S}_i \subseteq \mathcal{V}_i$ for each agent. Moreover, we can describe the entire Markov chain over $\mathcal{S}$ as the Kronecker product of the collection of the Markov chains of each agent, i.e., $P = \bigotimes_{i\in\mathcal{M}} P_i$. Since $P_i \in \mathbb{M}^*_{\pi_i}$ for every $i \in \mathcal{M}$ the stationary distribution of P is $\pi = \bigotimes_{i\in\mathcal{M}} \pi_i$.

Next, we will extend the definition of average hitting time for Markov chains over $\mathcal{S}$. Let us define the set $\mathcal{S}_\mathcal{A} := \{(s_0, \ldots, s_{m-1}) \in \mathcal{S} \mid \exists\ i : s_i \in \mathcal{A}\}$ as the set indicating the states of the multi-agent Markov chain P where there is at least one patroller in a location in $\mathcal{A}$. Therefore, $m_{\mathcal{S}_\mathcal{A}}(P)$, as in Equation (3), characterizes how fast multiple agents can cover the locations in $\mathcal{A}$. This allows us to formulate an optimization problem that minimizes $m_{\mathcal{S}_\mathcal{A}}(P)$.

$$\begin{aligned} \min_{P\in\mathbb{M}^*_\pi} \quad & \pi^\top \left(I_{n^m} - E_{\mathcal{S}_\mathcal{A}} P E_{\mathcal{S}_\mathcal{A}}\right)^{-1} \delta_{\mathcal{S}_\mathcal{A}} \\ \text{subject to} \quad & P = P_0 \otimes \ldots \otimes P_{m-1} \\ & P_i \text{ is supported on } \mathcal{G}_i \quad \forall\ i \in \mathcal{M} \\ \text{with} \quad & \pi = \pi_0 \otimes \ldots \otimes \pi_{m-1} \end{aligned} \tag{7}$$

Incorporating decoupled Markov chains turns the optimization problem shown in Equation (7) into a non-convex program with size equal to $O(n^m)$. Nevertheless, this structure naturally suggests the use of distributed algorithms to minimize $m_{\mathcal{S}_\mathcal{A}}(P)$.

Instead of optimizing all the Markov chains simultaneously we propose an algorithm where each patroller i improves its own Markov chain P_i by minimizing surrogate functions of the multi-agent average hitting time $u_i(P_i \mid P_{-i})$ where all the other Markov chains P_{-i} are fixed as shown in Algorithm 1. The most natural choice for a surrogate

Algorithm 1: Distr. Avg. Hitting Time Minimization

```
Initialize P_i^(0) ∈ M*_{π_i} for all i and k ← 0.
repeat
    foreach i ∈ M do
        P_i^(k+1) ← arg min_{P_i} u_i (P_i | P_{-i}^(k))
    k ← k + 1
until convergence condition
```

function is just to define $u_i(P_i|P_{-i}) = m_{\mathcal{S}_\mathcal{A}}(P)$. However, evaluating $m_{\mathcal{S}_\mathcal{A}}(P)$ is intractable, even if the other agents Markov chains are fixed. Instead, we focus on the family of surrogate functions,

$$\begin{aligned} u_i\left(P_i|P_{-i}\right) &= \alpha_{\mathcal{A},i} \sum_{r=0}^{\infty} \pi_i^\top \left(\beta_{\mathcal{A},i} E_{\mathcal{A},i} P_i E_{\mathcal{A},i}\right)^r \delta_{\mathcal{A},i} \\ &= \alpha_{\mathcal{A},i} \pi_i^\top \left(I_n - \beta_{\mathcal{A},i} E_{\mathcal{A},i} P_i E_{\mathcal{A},i}\right)^{-1} \delta_{\mathcal{A},i} \end{aligned} \tag{8}$$

where,

$$\alpha_{\mathcal{A},i} = \prod_{j\neq i} ||\delta_{\mathcal{A},j}||^2, \quad \text{and} \quad \beta_{\mathcal{A},i} = \prod_{j\neq i} \lambda_{\max}\left(E_{\mathcal{A},j} P_j E_{\mathcal{A},j}\right).$$

With the proposed choice of surrogate functions we recover multiple desirable properties that we lost in the multi-agent patrolling formulation. First, for each agent, the function $u_i(P_i \mid P_{-i})$ is a convex function of P_i. In addition, we reduce the size of the problem from $O(n^m)$ to m different $O(n)$ optimization problems per iteration. Moreover, given that each agent just need the values of $\alpha_{\mathcal{A},i}$ and $\beta_{\mathcal{A},i}$ from the other agents, instead of their transition matrices P_{-i}, the Algorithm 1 can be solved distributively. Finally, we can offer convergence guarantees for our proposed solution.

Theorem 4.1: Convergence of Distributed Average Hitting Time Minimization. For any non-empty subset of locations $\mathcal{A}$, every limit point $P^* = \left(P^*_0, \ldots, P^*_{m-1}\right)$ generated by Algorithm 1 is a coordinate minimum of $m_{\mathcal{S}_\mathcal{A}}(P)$.

Now, we devote the rest of this section to prove the statement in Theorem 4.1.

A. Preliminaries on Block Optimization

Consider the optimization problem $\min_{x\in\mathcal{X}} f(x)$, where $f : \mathcal{X} \to \mathbb{R}$ and $\mathcal{X}$ is a closed convex set. Now, assume that the variable x can be decomposed as $(x_0, \ldots, x_m) \in \mathcal{X}$ where $x_i \in \mathcal{X}_i$. Then, the update rule at iteration k for each block can be computed by the subproblem,

$$x_i^{(k+1)} \in \arg\min_{x_i\in\mathcal{X}_i} \quad u_i\left(x_i \mid x^{(k)}\right), \tag{9}$$

where $u_i : \mathcal{X}_i \times \mathcal{X} \to \mathbb{R}$. Convergence of the sequence generated by Equation (9) is guaranteed under the following properties on the functions u_i,

Upper Boundedness: $u_i(x_i|y) \geq f(y_1, \ldots, x_i, \ldots, y_m)$ for all $x_i \in \mathcal{X}_i$ and $y \in \mathcal{X}$.

Tightness: $u_i(x_i|x) = f(x)$ for all $x_i \in \mathcal{X}_i$ and $x \in \mathcal{X}$.

Continuity and Differentiability: $u(x_i|y)$ is continuous in (x_i, y) and differentiable on x_i.

Every function u_i that have these properties is called a *surrogate function* of $f(x)$ for the block of variables x_i. Thus, algorithm described by the update in Equation (9) is called Block Successive Upper-bound Minimization (BSUM) algorithm and its convergence to a coordinatewise minimum have been studied in [15, Thm. 2(a)].

B. Proof of Convergence of the Distributed Average Hitting Time Minimization Algorithm

Now, we prove convergence of Algorithm 1 using convergence of BSUM algorithm and establishing that minimizing functions $u_i(P_i \mid P_{-i})$ is equivalent to minimize surrogate functions of $m_{\mathcal{S}_\mathcal{A}}(P)$ for block variable P_i. With this in mind, we redefine the expression for $m_{\mathcal{S}_\mathcal{A}}(P)$ presented in Equation (7). Let us define $\delta_{\mathcal{A},l}(i) = \delta_\mathcal{A}(i)$ for each agent $l \in \mathcal{M}$. Therefore, we can define $\delta_{\mathcal{S}_\mathcal{A}} = \bigotimes_{l\in\mathcal{M}} \delta_{\mathcal{A},l}$ and $E_{\mathcal{S}_\mathcal{A}} = \bigotimes_{l\in\mathcal{M}} E_{\mathcal{A},l}$, with $E_{\mathcal{A},l} = \operatorname{diag}(\delta_{\mathcal{A},l})$. This allows us to rewrite the expression of average hitting time as,

$$m_{\mathcal{S}_\mathcal{A}}(P) = \sum_{r=0}^{\infty} \left[\prod_{l=0}^{m-1} \pi_l^\top \left(E_{\mathcal{A},l} P_l E_{\mathcal{A},l}\right)^r \delta_{\mathcal{A},l} \right] \tag{10}$$

Equation (10) shows that the coupling between the different agents in $m_{\mathcal{S}_\mathcal{A}}(P)$ appears as the bilinear forms $\pi_l^\top \left(E_{\mathcal{A},l} P_l E_{\mathcal{A},l}\right)^r \delta_{\mathcal{A},l}$ for any $r \geq 0$. Therefore, each patroller can upper bound Equation (10) if the other patrollers fix their Markov chains.

Lemma 4.2: Distributed Upper Bound of Multi-Agent Average Hitting Time. For any agent i, the functions $u_i\left(P_i|P_{-i}\right)$, as stated in Equation (8), are,

- an upper bound of $m_{\mathcal{S}_\mathcal{A}}(P_i, P_{-i})$ for any given P_{-i}.
- differentiable a.e. for both P_i and P_{-i}.
- a convex function on P_i.

Proof: In order to check the upper bound property of Equation (8), we only need to upper bound each of the bilinear forms $\pi_i^\top \left(E_{\mathcal{A},i} P_i E_{\mathcal{A},i}\right)^r \delta_{\mathcal{A},i}$ for any $i \in \mathcal{M}$. Thus,

$$\begin{aligned}\pi_i^\top \left(E_{\mathcal{A},i} P_i E_{\mathcal{A},i}\right)^r \delta_{\mathcal{A},i} &= \pi_i^\top E_{\mathcal{A},i}\left(E_{\mathcal{A},i} P_i E_{\mathcal{A},i}\right)^r \delta_{\mathcal{A},i} \\ \leq 1^\top E_{\mathcal{A},i}\left(E_{\mathcal{A},i} P_i E_{\mathcal{A},i}\right)^r \delta_{\mathcal{A},i} &\leq \lambda_{\max}\left(E_{\mathcal{A},i} P_i E_{\mathcal{A},i}\right)^r ||\delta_{\mathcal{A},i}||^2\end{aligned}$$

Then, we can upper bound $\prod_{j\neq i} \pi_j^\top \left(E_{\mathcal{A},j} P_j E_{\mathcal{A},j}\right)^r \delta_{\mathcal{A},j}$ by $\alpha_{\mathcal{A},i}\beta_{\mathcal{A},i}^r$ to obtain the expression in Equation (8).

To check the differentiabilty of the upper bound we just need to verify if $\lambda_{\max}\left(E_{\mathcal{A},i} P_i E_{\mathcal{A},i}\right)$ is a differentiable function of P_i for all i. For any symmetric matrix A, each of the eigenvalues $\lambda_p(A)$ is a Lipschitz continuous function of A [16]. Then, by Rademacher's theorem, the function $\lambda_{\max}(A)$ is differentiable a.e. in its domain. Note that, even when Markov chains P_i are not symmetric, due to the reversibility property they are similar to the symmetric matrix $Q_i = D_i^{\frac{1}{2}} P_i D_i^{-\frac{1}{2}}$ with $D_i = \operatorname{diag}(\pi_i)$. Finally, convexity of Equation (8) comes form convexity of $m_\mathcal{A}(P)$ and the fact that $\beta_{\mathcal{A},i} \leq 1$ and $\lambda_{\max}\left(E_{\mathcal{A},j} P_j E_{\mathcal{A},j}\right) \leq \rho\left(E_{\mathcal{A},j} P_j E_{\mathcal{A},j}\right) < 1$ for any $j \neq i$. ■

Lemma 4.2 verifies two of the required properties of the surrogate functions. Although, it is possible that the upper bound presented in Equation (8) is not tight. However, we just need to guarantee that minimizing $u_i(P_i \mid P_{-i})$ is equivalent to minimizing a surrogate function, even if u_i is not a surrogate function for the block of variables P_i.

Proposition 4.3: For any $\mathcal{A}$, the family of functions,

$$\begin{aligned} g_i\left(P_i|P_i^{(k)}, P_{-i}^{(k)}\right) = u_i\left(P_i|P_{-i}^{(k)}\right) + m_{\mathcal{S}_\mathcal{A}}\left(P_i^{(k)}, P_{-i}^{(k)}\right) \\ - u_i\left(P_i^{(k)}|P_{-i}^{(k)}\right) \end{aligned} \tag{11}$$

are surrogate functions of $m_{\mathcal{S}_\mathcal{A}}(P)$ for every i. Moreover,

$$\arg\min_{P_i} g_i\left(P_i|P_i^{(k)}, P_{-i}^{(k)}\right) = \arg\min_{P_i} u_i\left(P_i|P_{-i}^{(k)}\right). \tag{12}$$

Proof: Tightness of g_i can be checked by inspection of Equation (11). Similarly, continuity and differentiability of g_i comes from continuity and differentiability of $u_i\left(P_i|P_{-i}\right)$, verified in Lemma 4.2. Relation in Equation (12) can be verified giving the fact that the last two terms in Equation (11) do not depend on P_i. In order to verify that g_i is an upper bound of $m_{\mathcal{S}_\mathcal{A}}(P)$ we just need to verify that $u_i\left(P_i|P_{-i}^{(k)}\right) - u_i\left(P_i^{(k)}|P_{-i}^{(k)}\right) \geq m_{\mathcal{S}_\mathcal{A}}\left(P_i, P_{-i}^{(k)}\right) - m_{\mathcal{S}_\mathcal{A}}\left(P_i^{(k)}, P_{-i}^{(k)}\right)$. Using Equations (10) and (8), we have that,

$$\begin{aligned}&\sum_{r=0}^{\infty} \alpha_{\mathcal{A},i}\beta_{\mathcal{A},i}^r \pi_i^\top \left[\left(E_{\mathcal{A},i} P_i E_{\mathcal{A},i}\right)^r - \left(E_{\mathcal{A},i} P_i^{(k)} E_{\mathcal{A},i}\right)^r\right] \delta_{\mathcal{A},i} \\ &\geq \sum_{r=0}^{\infty} \left[\prod_{j\neq i} \pi_j^\top \left(E_{\mathcal{A},j} P_j E_{\mathcal{A},j}\right)^r \delta_{\mathcal{A},j}\right] \pi_i^\top \left[\left(E_{\mathcal{A},i} P_i E_{\mathcal{A},i}\right)^r - \left(E_{\mathcal{A},i} P_i^{(k)} E_{\mathcal{A},i}\right)^r\right] \delta_{\mathcal{A},i}\end{aligned}$$

Therefore, for any r such that $\left(E_{\mathcal{A},i} P_i E_{\mathcal{A},i}\right)^r \neq \left(E_{\mathcal{A},i} P_i^{(k)} E_{\mathcal{A},i}\right)^r$ the upper bound inequality is guaranteed by Lemma 4.2. ■

Given that Proposition 4.3 shows that Algorithm 1 is equivalent to iteratively minimize surrogate functions of $m_{\mathcal{S}_\mathcal{A}}(P)$ we can guarantee the convergence of Algorithm 1, as stated in Theorem 4.1.

V. Numerical Simulations

In this section we present numerical simulations to illustrate the usefulness of our proposed solution. We implement our strategy in two different road networks and compare our solutions with other baseline strategies that design Markov chains with a predefined stationary distribution. As a baseline algorithm to generate Markov chains supported in a graph $\mathcal{G}$ we use the Metropolis-Hasting algorithm. In it, the transitions are defined as $p_{ij} = \frac{1}{r}\min\left(1, \frac{\pi(j)}{\pi(i)}\right)$ for $i \neq j$ and $p_{ii} = 1 - \sum_j p_{ij}$, where r is the maximum degree for all nodes in $\mathcal{G}$. It is known that the generated Markov chain is reversible and has stationary distribution π [17, Chap. 4].

Case Study 1: Map of San Francisco. In the first set of simulations we implement our algorithm in a data set consisting of 12 locations in San Francisco and their crime rates. Using the data in [18] we identify 12 locations on San Francisco to be monitored persistently. With the addition of intermediate nodes, we build a planar graph that describes how the agents can move through the city as shown in Figure 1a. Then, we create 5 different subgraphs that represent the assigned locations for each one of the agents as in Figure 1b. Given the overlaps between the different subgraphs, every agent has to take into consideration other agents' strategies to minimize the worst-case average hitting

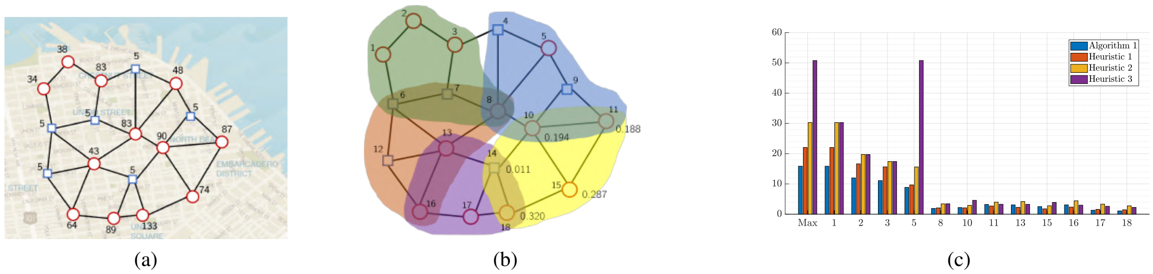

Fig. 1. San Francisco case study. (a) Locations that represent the different states of the Markov chains and their respective crime rates. The red circles are the 12 critical locations reported in [18] while the blue squares represent intermediate locations. (b) Labels for the nodes and the different subgraphs assigned for each one of the 5 different agents and an example of the stationary distribution for one of the subgraphs. (c) Average hitting time to each one of the target locations using the different optimization algorithms.

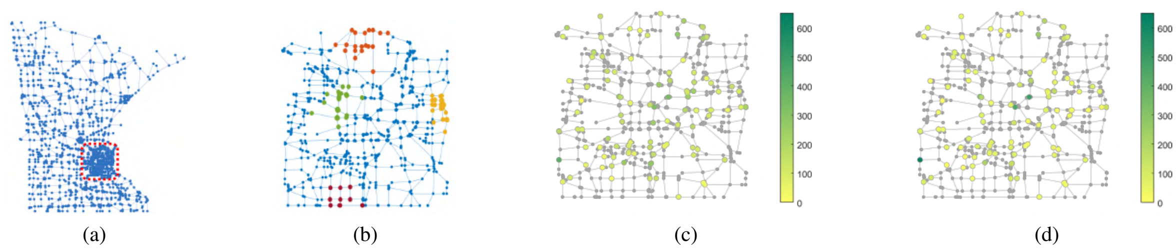

Fig. 2. Minneapolis case study. (a) Minnesota road network obtained from [19]. The most dense region is highlighted inside the red dashed square. (b) Network used in the surveillance problem that corresponds to the road network of Minneapolis and surroundings. Example of 4 of the subgraphs used in the surveillance design highlighted using different colors on the nodes. (c) Obtained average hitting times for every target location using Algorithm 1. (d) Obtained average hitting times for every target location using Heuristic 1.

time in the 12 critical locations, as in Equation (5). Moreover, we define the stationary distributions for each subgraph such that they are proportional to the crime rates provided in [18] and inversely proportional to the number of patrollers that cover it, as shown in Figure 1b.

Under this set-up we obtain a problem with $n = 18$ locations and $m = 5$ agents. Therefore, calculating the optimal surveillance strategy will require a Markov chain with 4500 states. However, using one iteration of our proposed algorithm just require that each agent optimize their own Markov chain, reducing it to 6 states in the worst case, giving a total size of 28 states per iteration.

Since using centralized algorithms is unreliable, we implement other three reasonable heuristics that can be computed distributively to compare with our surveillance strategy. Those strategies take each subgraph $\mathcal{G}_i$ and,
Heuristic 1: Find the Markov chain P_i that minimize the average hitting time, neglecting the presence of other agents.
Heuristic 2: Apply the Metropolis-Hasting algorithm using π_i as stationary distribution.
Heuristic 3: Apply the Metropolis-Hasting algorithm over a tour contained in $\mathcal{G}_i$ using π_i as stationary distribution.

In Figure 1c we can observe the achieved hitting time for every target location using the different approaches. Comparing the two heuristics based upon the Metropolis-Hasting algorithm it can be noticed that constraining the agent to move just over a tour instead of the entire graph significantly increase the hitting time to certain locations. Instead, when each agent minimizes the average hitting time, we can observe an improvement from the Metropolis-Hasting algorithm. The improvement is due we are choosing a Markov chain that not only has a particular stationary distribution but also minimizes the average hitting time for each subgraph. However, ignoring the existing overlaps on the subgraphs leads to unnecessary visit for multiple agents. On the other hand, incorporating the effect of the coupling of the patrolling schemes in our design, as in our proposed algorithm, leads to a better performance overall, achieving lower hitting times for multiple locations.

Case Study 2: Map of Minneapolis. In a similar fashion of previous case study, now we apply the proposed algorithm in the road network of Minneapolis. To obtain this graph we start with the Minnesota road network as provided in [19]. However, we focus on the more dense region as shown in Figure 2a. To check the scalability of our proposed optimization problem, we design a patrolling scheme that consider 50 patrolling agents. With this in mind, we generate the different subgraphs for each agent using spectral decomposition over the adjoint graph. This decomposition features some overlap for the subgraphs, letting multiple agents cover some of the nodes. Some of the subgraphs generated are shown in Figure 2b. To define the stationary distributions we just impose the constraint that they should be proportional to the degree of each node, i.e. the number of edges that are connected to that node. The target nodes are randomly chosen with probability proportional to a centrality measure, in particular the PageRank centrality. Similarly to the previous set of simulations, our goal is to minimize the worst-case average hitting time for the target nodes.

Recall that, for n locations and m agents, a Markov chain with $O(n^m)$ states is required to minimize the average hitting time in a centralized manner. Instead, one iteration of our algorithm requires m Markov chains with $O(n)$ states. For this case study we have $n = 774$ locations, $|\mathcal{E}| = 963$ edges and $m = 50$ agents. However, the dimension of the subgraphs, ranging between 8 and 38, are smaller compared to the total number of nodes. Thus, the Distributed Average Hitting Time Minimization presented in Algorithm 1 leads to a huge dimensionality reduction of the surveillance design problem compared to a centralized algorithm, which approximately requires 4×10^{62} states.

As for previous case study, we compare our strategy with another baseline algorithm to generate Markov chains. In particular, given the previously discussed results, we compare only with the Markov chains generated by Heuristics 1 & 2. In Figures 2c and 2d it can be noticed that our proposed algorithm achieves a better performance in terms of the worst-case average hitting time. Again, the reason of this behavior is due the fact that our algorithm takes into account the interaction between the agents. Keeping this interactions during the optimization problem allow us to design patrolling strategies that exploit the overlaps present along the different subgraphs. Also, as we showed in our discussion, using our solution we can include this interaction without a significant increase in the complexity of the optimization problem.

TABLE I

SUMMARY OF OBTAINED AVERAGE HITTING TIMES.

Hitting Time Minimization	Case Study 1: San Francisco		Case Study 2: Minneapolis	
	Worst Case	Mean	**Worst Case**	Mean
Algorithm 1	**15.8533**	**6.3158**	**417.2229**	132.9839
Heuristic 1	22.0511	7.8801	646.4553	**88.4718**
Heuristic 2	30.2835	10.8595	501.1369	139.7152
Heuristic 3	50.7946	15.0315	-	-

In Table I we summarize the results obtained for both case studies. In it, we can compare the performance of our strategy and the aforementioned heuristics. It is evident that our solution outperform other strategies in terms of the desired metric: the worst-case average hitting time. Moreover, we observe that our solution also inherently reduces the average hitting for multiple target nodes, achieving similar or even better performance in terms of the mean average hitting time.

VI. CONCLUSIONS

In this work we present an algorithm that provides a patrolling scheme for a multi-agent robotic surveillance problem. By using the Markov chains mathematical framework we formulate an optimization problem for the patrolling strategy that covers all the relevant locations as fast as possible. Under mild assumptions on the behavior of the agents we can extend this formulation to environments with multiple agents. Moreover, we design easy to implement and highly unpredictable surveillance strategies by constraining ourselves to decoupled reversible Markov chains. Using properties of decoupled reversible Markov chains, we were able to define a distributed algorithm that iteratively refines the surveillance strategy used by all the robotic agents. In addition, we provide sufficient conditions to ensure the convergence of the proposed algorithm. With a distributed algorithm, we guarantee the convergence to a coordinate minimum of the multi-agent average hitting time while simultaneously improving the tractability of the resulting optimization problems. Through numerical simulations, we verify the performance of our solution, showing its capabilities even for large scale formulations.

Future directions include the extension of these algorithms for arbitrary travel times between locations and the design of the parameters used in our algorithm, including the subgraphs for each agent and their stationary distributions.

REFERENCES

[1] X. Duan and F. Bullo, "Markov Chain-based Stochastic Strategies for Robotic Surveillance," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, pp. 243–264, 2021.

[2] Y. Li, Y. Luo, and C. Yu, "Game-Theoretic Control of Markov Chains: Two Applications," in *International Symposium on Mathematical Theory of Networks and Systems*, 2022.

[3] N. Agmon, G. A. Kaminka, and S. Kraus, "Multi-robot Adversarial Patrolling: Facing a Full-knowledge Opponent," *Journal of Artificial Intelligence Research*, vol. 42, pp. 887–916, 2011.

[4] N. Basilico, N. Gatti, and F. Amigoni, "Patrolling Security Games: Definition and Algorithms for Solving Large Instances with Single Patroller and Single Intruder," *Artificial Intelligence*, vol. 184, pp. 78–123, 2012.

[5] A. B. Asghar and S. L. Smith, "A Patrolling Game for Adversaries with Limited Observation Time," in *IEEE Conference on Decision and Control*, 2018, pp. 3305–3310.

[6] Y. Vorobeychik, B. An, and M. Tambe, "Adversarial Patrolling Games," in *AAAI Spring Symposium Series*, 2012.

[7] X. Duan, D. Paccagnan, and F. Bullo, "Stochastic Strategies for Robotic Surveillance as Stackelberg Games," *IEEE Transactions on Control of Network Systems*, vol. 8, no. 2, pp. 769–780, 2021.

[8] Y. Chevaleyre, "Theoretical Analysis of the Multi-Agent Patrolling Problem," in *IEEE/WIC/ACM International Conference on Intelligent Agent Technology.*, 2004, pp. 302–308.

[9] R. Patel, P. Agharkar, and F. Bullo, "Robotic Surveillance and Markov Chains with Minimal Weighted Kemeny Constant," *IEEE Transactions on Automatic Control*, vol. 60, no. 12, pp. 3156–3167, 2015.

[10] M. George, S. Jafarpour, and F. Bullo, "Markov Chains with Maximum Entropy for Robotic Surveillance," *IEEE Transactions on Automatic Control*, vol. 64, no. 4, pp. 1566–1580, 2018.

[11] J. Grace and J. Baillieul, "Stochastic Strategies for Autonomous Robotic Surveillance," in *Proceedings of the 44th IEEE Conference on Decision and Control*, 2005, pp. 2200–2205.

[12] R. Patel, A. Carron, and F. Bullo, "The Hitting Time of Multiple Random Walks," *SIAM Journal on Matrix Analysis and Applications*, vol. 37, no. 3, pp. 933–954, 2016.

[13] J. R. Norris, *Markov Chains*. Cambridge University Press, 1998.

[14] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[15] M. Razaviyayn, M. Hong, and Z.-Q. Luo, "A Unified Convergence Analysis of Block Successive Minimization Methods for Nonsmooth Optimization," *SIAM Journal on Optimization*, vol. 23, no. 2, pp. 1126–1153, 2013.

[16] Y. Xu and Y. Lai, "Derivatives of Functions of Eigenvalues and Eigenvectors for Symmetric Matrices," *Journal of Mathematical Analysis and Applications*, vol. 444, no. 1, pp. 251–274, 2016.

[17] A. Blum, J. Hopcroft, and R. Kannan, "Foundations of Data Science," *Vorabversion eines Lehrbuchs*, vol. 5, no. 5, 2016.

[18] S. Alamdari, E. Fata, and S. L. Smith, "Persistent Monitoring in Discrete Environments: Minimizing the Maximum Weighted Latency between Observations," *International Journal of Robotics Research*, vol. 33, no. 1, pp. 138–154, 2014.

[19] T. A. Davis and Y. Hu, "The University of Florida Sparse Matrix Collection," *ACM Transactions on Mathematical Software*, vol. 38, no. 1, pp. 1–25, 2011.