Persistent Monitoring of Events with Stochastic Arrivals at Multiple Stations

Jingjin Yu[1,2]   Sertac Karaman[3]   Daniela Rus[1]

Abstract—This paper is concerned with a novel mobile sensor scheduling problem, involving a single robot tasked with monitoring several events of interest that occur at different locations. Of particular interest is the monitoring of events that can not be easily forecast. Prominent examples range from natural phenomena (*e.g.*, monitoring abnormal seismic activity around a volcano using a ground robot) to urban activities (*e.g.*, monitoring early formations of traffic congestion in the Boston area using an aerial robot). Motivated by these examples, this paper focuses on problems where the precise occurrence time of the events is not known *a priori*, but some statistics for their inter-arrival times are available from past observations. The robot's task is to monitor the events to optimize the following two objectives: *(i)* maximize the number of events observed and *(ii)* minimize the delay between two consecutive observations of events occurring at the same location. Provided with only one robot, it is crucial to optimize these objectives in a balanced way, so that they are optimized at each station simultaneously. Our main theoretical result is that this complex mobile sensor scheduling problem can be reduced to a quasi-convex program, which can be solved in polynomial time. In other words, a globally optimal solution can be computed in time that is polynomial in the number of locations. We also provide computational experiments that validate our theoretical results.

I. INTRODUCTION

Consider a single robotic vehicle that is tasked with monitoring events that occur at several locations. Unfortunately, the precise occurrence time of an event is unknown to the robot *a priori*. Hence, the robot must travel to the particular location and wait for the event to occur, in order to monitor the event and capture the data associated with it. Ideally, one would like to monitor all events at all locations. However, provided with a single robot, one must optimize the schedule of the robot to ensure that all locations are observed equally well as best as possible, *i.e.*, in a balanced manner. Two major objectives are to *(i)* ensure that a large number of events are observed at each location and *(ii)* ensure that the delay between two observations of events at any given location is minimized. Optimizing these objectives in a balanced manner is a fairly complex, multi-objective scheduling problem.

The problem setup we study in this paper is novel, and it is applicable to a broad set of applications concerning persistent data collection through monitoring a set of events
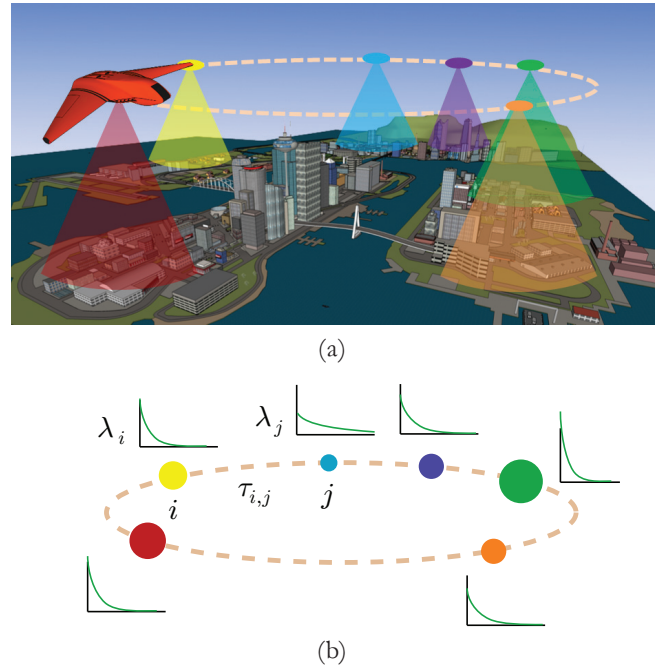

(a)

(b)

Fig. 1. (a) One of many potential applications of our persistent monitoring formulation, in which an UAV (robot) is given the task of continuously gathering stochastically occurring (data) events at a set of fixed locations (the surface areas under the colored cones). The sizes of the colored discs represent the stochastic arrival rates of events at the locations. (b) Illustration of the underlying geometric problem setting. At each point of interest, say location (station) i, events arrive following a Poisson process with intensity λ_i. It takes a robot $\tau_{i,j}$ time to move from station i to station j, during which no observation can be made. The associated plots roughly capture the (exponential) distributions of event arrivals associated with the stations.

at various locations. The events of interest include natural phenomena (e.g., volcanic eruptions and early formations of blizzards, hailstorms, and tsunamis), biological disasters (e.g., early formations of epidemic diseases on animal or plant populations), as well as military operations (e.g., terrorist attacks). The key common characteristic of these events is that their precise time of occurrence can not be easily forecast, although the statistics regarding how often they occur may be available from past experience. Hence, the data-collecting robot must wait at the location of interest to capture the event once it occurs. Then, the fundamental scheduling problem is to decide how much time the robot should spend in each location to archive various objectives, such as those described above, in a balanced way. Our main theoretical result is that this complex multi-objective mobile sensor scheduling problem can be reduced to a strictly quasi-convex optimization problem

that can be solved in polynomial time. Hence, the (unique) globally optimal solution of this complex scheduling problem can be computed in time that is polynomial in the number of locations.

Broadly speaking, persistent monitoring problems appear naturally whenever only limited resources are available for serving a set of spatially-dispersed tasks. Motivated by a variety of potential applications [1], [2], several authors have studied persistent monitoring problems [3]–[11]. For example, in [3], the authors consider a certain weighted latency measure as a robot continuously traverse a graph, in which the vertices represent the regions of interest and the edges between the vertices are labeled with the travel time. They show that the problem of minimizing the maximum latency across all stations is computationally intractable, and they present an approximation algorithm. In [5], the authors consider a persistent monitoring problem for a group of agents in a one-dimensional mission space. They show that this problem can be solved by parametrically optimizing a sequence of switching locations for the agents. The problem of generating speed profiles for robots along predetermined closed paths for keeping bounded a varying field is addressed in [10]. The authors characterize appropriate policies for both single and multiple robots. In [11], decentralized adaptive controllers were designed to morph the initial closed paths of robots to focus on regions of high importance.

In contrast to all the references cited above, the problem studied in this paper focuses on transient events, emphasizing unknown arrival times (but known statistics). The event arrival times being unknown forces the robot to wait at each station in order to observe the events of interest.

Persistent surveillance problems are intimately linked with coverage problems. Coverage of a two-dimensional region has been extensively studied in robotics [12]–[14], as well as in purely geometric settings, for example, in [15], where the proposed algorithms compute the shortest closed routes for continuous coverage of polygonal interiors under an infinite visibility sensing model. Coverage with limited sensing range was also addressed later [16], [17]. If the environment to be monitored has a 1-dimensional structure, discrete optimization problems, such as the Traveling Salesman Problem, often arise [3]. In most coverage problems, including those cited above, the objective is to place sensors in order to maximize, for example, the area that is within their sensing region. The persistent surveillance problem we study in this paper is a special case, where the limited number of sensors do not allow extensive coverage; hence, we resort to mobility in order to optimize the aforementioned performance metrics.

Persistent monitoring problems are also related to (static) *sensor scheduling* problems (see, e.g., [18]–[20]), which are usually concerned with scheduling the activation times of sensors in order to maximize the information collected about a time-varying process. The problem considered in this paper involves a *mobile sensor* that can travel to each of the locations, where the additional time required to travel between stations is non-zero. The mobile sensor scheduling literature is also rich.

For instance, in [21], the authors study the control of a robotic vehicle in order to maximize data rate while collecting data stochastically arriving at two locations. The problem studied in this paper is a novel mobile sensor scheduling problem involving several locations and a multi-objective performance metric that includes both the data rate and the delay between consecutive observations.

The main contributions of this paper are two-fold. First, we propose a novel persistent monitoring and data collection problem, with the unique feature that the precise arrival times of events are unknown *a priori*, but their statistics are available. Modeling the arrival of events as a stochastic process allows our formulation to encompass several practical applications, where the precise occurrence times of the events of interest can not forecast easily. Second, focusing on cyclic policies, we establish that this fairly complex multi-objective mobile sensor scheduling problem admits a globally optimal solution that can be computed efficiently in polynomial time. Surprisingly, it can be shown that the main objective is quasi-convex on its entire domain, which greatly simplifies the computation for finding the extremal values.

The rest of the paper is organized as follows. A precise definition of our persistent monitoring problem is provided in Section II. We then carry out the analysis and present our main result in detail in Section III, followed by experimental validation through simulation in Section IV. We conclude with Section V.

II. PROBLEM STATEMENT

Before formulating the problem, for convenience, we list several frequently used symbols and their meanings in Table I. When in doubt, the reader is referred to this table.

We study the problem of using a single robot to monitor events that occur at different stations. The robot can monitor one station at a time. It can travel from one station to another if the two stations are topologically connected. The precise time that an event will occur is not known to the robot *a priori*. However, the robot is provided with their statistics, for example the inter-arrival times, for each station. The robot can observe an event generated by a station if and only if it is at the same station at the time of occurrence, in which case the robot collects valuable data regarding that particular station. Roughly speaking, our objective is to design a scheduling policy for the robot to ensure that:

- *Objective.(i)* maximize the number of events that is observed at each station in a balanced way;
- *Objective.(ii)* minimize the delay between consecutive observations at a particular station for all stations.

Below, this problem is formulated as a multi-objective optimization problem. In Section III, it is shown that an important special case, involving a chain of stations, can be reduced to a quasi-convex program that can be solved efficiently. Hence, the running time of the algorithm that solves this special case is polynomial in the number of stations.

A more formal description of the problem is the following. Consider a network of n stations, represented by a connected

v_i, k_i	Stations to be monitored
λ_i	Intensity of the Poisson process at station i
$\tau_{i,j}$	Travel time from station i to station j
π	Cyclic policy of the form $((k_1, t_1), \ldots, (k_n, t_n))$, in which t_i is the time spent by the robot at station i in one policy cycle
T	Total time incurred by a policy cycle
T_{tr}	Total travel time per policy cycle
T_{obs}	$T - T_{tr}$, total observation time per policy cycle
$N_i(\pi)$	The number of events collected at station i in one period of the policy π
$T_i(\pi)$	The time between two consecutive event observations at station i containing travel to other stations, for the policy π
Π	$\arg\max_\pi \max_i \alpha_i(\pi)$
$p(X)$	Probability density of a random variable X
$Pr(e)$	Probability of an event e
$\mathbb{E}[X]$	Expected value of a random variable X
$\alpha_i(\pi)$	$\mathbb{E}[N_i(\pi)] / \sum_{j=1}^n \mathbb{E}[N_j(\pi)]$

graph $G = (V, E)$, where $V = \{v_1, v_2, \ldots, v_n\}$ is the set of vertices and E is the set of (directed) edges. If there exists an edge $(v_i, v_j) \in E$ between vertices v_i and v_j, then stations i and j are connected, meaning that the robot can travel from station i to station j directly. The time it takes the robot to travel from station i to station j is denoted by $\tau_{i,j}$.

Station i generates events at random time instances. More precisely, we model the arrival of events at station i with a Poisson process of intensity λ_i. These statistics, that is, the arrival processes being Poisson and their intensities being λ_i, are all known to the robot *a priori*, although the precise arrival times are not known beforehand.

A problem instance is fully characterized by the following parameters: *(i)* the graph, $G = (V, E)$, that represents the network of stations; *(ii)* the travel time, $\tau_{i,j}$, from station i and j for all i, j with $(v_i, v_j) \in E$; *(iii)* the arrival rates of the events, λ_i, for each station i. Given such a problem instance, we would like to design a routing policy for the robot to visit each station and spend a certain amount of time, in order to collect data through observing events so as to optimize the objective function, which we roughly described above. Precise definitions of these objectives will follow shortly.

A *cyclic policy* is one that the robot visits each station in a fixed order and spends a fixed amount of time at each station. More precisely, a cyclic policy is fully characterized by: *(i)* an ordering of stations, say $k_1, k_2, \ldots, k_n$, where $k_i \in \{1, 2, \ldots, n\}$ and $k_i \neq k_j$ for all i, j, and *(ii)* the time spent at each station, say t_i time units at station i. Such a cyclic policy is executed by first visiting station k_1 to spend t_1 time units, then visiting station k_2 to spend t_2 time units, then visiting station k_3 to

spend t_3 time units, and so on. A cyclic policy, which we denote by π, can be represented by the parameters listed above, as in $\pi = ((k_1, t_1), (k_2, t_2), \ldots, (k_n, t_n))$. Throughout the paper, we consider only cyclic policies.

Given a cyclic policy $\pi = ((k_1, t_1), (k_2, t_2), \ldots, (k_n, t_n))$, we define the aforementioned two objectives as follows. Let $N_i(\pi)$ denote the number of events that are observed at station i during one cycle. Define the fraction of events observed at station i as $\alpha_i(\pi) := \mathbb{E}[N_i(\pi)]/\mathbb{E}[\sum_{j=1}^n N_j(\pi)] = \mathbb{E}[N_i(\pi)]/\sum_{j=1}^n \mathbb{E}[N_j(\pi)]$. To formalize the first objective, we consider selecting a policy π that maximizes the minimum fraction of events, where the minimum is taken across all stations, *i.e.*,

$$\max_\pi \min_i \alpha_i(\pi) = \max_\pi \min_i \frac{\mathbb{E}[N_i(\pi)]}{\sum_{j=1}^n \mathbb{E}[N_j(\pi)]}. \tag{1}$$

This objective function maximizes the fraction of events observed at each station.[1] It does so in a balanced manner, maximizing the minimum α_i across all stations.

We formalize the second objective as follows. Suppose the cyclic policy is run until time t_{start} such that *(i)* at least one event is observed at each station up until time t_{start} and *(ii)* the robot is at the beginning of a new cycle at time t_{start}. For each station i, define $T_i(\pi)$ as the time between the following two observations: *(i)* the last event that was recorded at station i before t_{start} and *(ii)* the first event that is recorded at station i after t_{start}. In essence, $T_i(\pi)$ is the delay between two consecutive observations that fall into different observation windows, at station i. Our objective is to minimize these delays, again in a balanced manner across all stations. Hence, we consider choosing a policy that minimizes the maximum delay across all stations, i.e.,

$$\min_\pi \max_i \mathbb{E}[T_i(\pi)]. \tag{2}$$

In most cases, both objectives are equally important. One would like to maximize both the fraction of observations and minimize delays between observations, in some balanced manner across stations. Interestingly, the set of policies that optimize the first objective function is not unique; in fact, there are infinitely many such cyclic policies. We compute the optimal (cyclic) policy for the second objective function among those policies that optimize the first objective function. That is, we compute the (unique) policy $\pi^* = \arg\min_{\pi \in \Pi} \max_i \mathbb{E}[T_i(\pi)]$, where $\Pi := \arg\max_{\pi'} \min_i \alpha_i(\pi')$. Below, we prove that Π is an uncountably infinite set of cyclic policies and π^* is unique.

III. THE OPTIMAL SCHEDULING ALGORITHM AND ITS ANALYSIS

In this section, we provide a cyclic routing policy (algorithm) that solves the problem described in the previous section. We prove that the proposed policy is optimal. First, we show (via Lemma 1) that for any fixed time period T, there

---

[1]Ideally, we would like to define the notion of expected fraction of events observed at station i as follows: $\mathbb{E}[N_i(\pi)/\sum_{j=1}^n N_j(\pi)]$. However, the random variable $N_i(\pi)/\sum_{j=1}^n N_j(\pi)$ is not well defined, as its denominator may be zero. Instead, we use the well-defined expression $\mathbb{E}[N_i(\pi)]/\sum_{j=1}^n \mathbb{E}[N_j(\pi)]$.

is a unique cyclic policy π that optimizes the first objective (Equation (1)). However, the optimal policies for different T's assign the same value to Equation (1), giving rise to a continuum of solutions for the first objective. This issue is resolved by our main theorem (Theorem 2), which shows that there is a unique T that optimizes the second objective (Equation (2)).

Throughout this section, we consider an important special case where the locations are connected in a "closed chain" configuration. That is, we consider the network of locations represented by the graph $G = (V, E)$ where the vertex set is $V = \{v_1, v_2, \ldots, v_n\}$ and the edge set E is such that $(v_i, v_{i+1}) \in E$ for all $i \in \{1, 2, \ldots, n-1\}$ and that $(v_n, v_1) \in E$. In this case, the locations form a closed chain, hence the robot must visit the locations in a fixed order. Our main result (Theorem 2) applies to this important case. We conjecture an important generalization of our result in Section V.

Let us consider the first objective function only and momentarily ignore the second objective function. Then, the following lemma characterizes the set of all policies that optimize the first objective function, which was given by Equation (1). Since we consider only cyclic policies, the travel time T_{tr} for the robot per cyclic period is fixed:

$$T_{\mathrm{tr}} = \sum_{i,j} \tau_{i,j} \qquad 1 \leq i \leq n, j = (i+1) \mod n. \qquad (3)$$

For notational convenience, we define $\gamma_i := 1/(\lambda_i \sum_{j=1}^{n} (1/\lambda_j))$.

Lemma 1 *Among all cyclic policies, a cyclic policy $\pi = ((k_1, t_1), \ldots, (k_n, t_n))$ optimizes the first objective function, i.e.,*

$$\pi \in \operatorname*{argmax}_{\pi'} \min_i \frac{\mathbb{E}[N_i(\pi')]}{\sum_{j=1}^{n} \mathbb{E}[N_j(\pi')]}$$

if and only if

$$t_i = \gamma_i (T - T_{\mathrm{tr}}), \qquad (4)$$

where $T = \sum_{i=1}^{n} t_i + T_{\mathrm{tr}}$ is a parameter (the cyclic policy's period). For $T > T_{\mathrm{tr}}$, the resulting cyclic policy optimizes the first objective. Moreover, such a cyclic policy π satisfies:

$$\mathbb{E}[N_1(\pi)] = \mathbb{E}[N_2(\pi)] = \cdots = \mathbb{E}[N_n(\pi)]. \qquad (5)$$

PROOF. Since we are looking at cyclic policies, by linearity of expectations, the value of the first objective, as defined in Equation (1), remains the same if we only look at a single policy cycle (versus looking at an infinite time horizon). We show that for arbitrary $T > T_{\mathrm{tr}}$, choosing t_i's according to Equation (4) yields the same optimal value for Equation (1). Now fixing a policy π, after spending t_i time at station i, the robot collects $\mathbb{E}[N_i(\pi)] = \lambda_i t_i$ data points in expectation. This yields

$$\alpha_i(\pi) = \frac{\mathbb{E}[N_i(\pi)]}{\sum_{j=1}^{n} \mathbb{E}[N_j(\pi)]} = \frac{\lambda_i t_i}{\sum_{j=1}^{n} \lambda_j t_j}.$$

It is straightforward to see that $\min_i \alpha_i(\pi)$ is maximized if and only if Equation (5) is satisfied, yielding a value of $1/n$ for Equation (1). Solving the equations $\lambda_1 t_1 = \ldots = \lambda_n t_n$ and

$\sum_{i=1}^{n} t_i = T - T_{\mathrm{tr}}$ together then yields Equation (4). $\square$

Lemma 1 has two important implications. Firstly, any cyclic policy that equalizes the expected number of events observed at each station optimizes the first objective function given by Equation (1). This provides us with an uncountably infinite set of optimal policies (optimal for the first objective function only), which is the second immediate implication of the lemma. Any cyclic policy that satisfies Equation (4) is optimal, independently of the value of T. Let us emphasize that Lemma 1 is particularly important since it characterizes the set of policies that optimize the first objective function given by Equation (1). Next, we show that, among those cyclic policies that optimize the first objective function, there exists a unique cyclic policy that optimizes the second objective (see Equation (2)). Moreover, this unique optimal policy can be computed by solving a quasi-convex optimization problem, which can be done efficiently in polynomial time.

Theorem 2 *Let Π denote the (uncountably infinite) set of cyclic policies that maximizes the first objective function given by Equation* (1)*, i.e.,*

$$\Pi := \operatorname*{argmax}_{\pi} \min_i \alpha_i(\pi). \qquad (6)$$

Then, there exists a unique cyclic policy in Π that minimizes the second objective function given by Equation (2)*. This policy is in the form given by Equation* (4)*, i.e.,*

$$t_i^* = \gamma_i (T^* - T_{\mathrm{tr}}) \qquad \text{for all } i, \qquad (7)$$

where

$$T^* := \operatorname*{argmin}_{T > T_{\mathrm{tr}}} \max_i \left[\frac{2}{\lambda_i} + \frac{(T - t_i)(1 + e^{-\lambda_i t_i})}{1 - e^{-\lambda_i t_i}} \right], \qquad (8)$$

which is a quasi-convex optimization problem, i.e., the objective function is quasi-convex in T. Hence, the optimal policy that solves the problem described in Section II can be computed efficiently in polynomial time.

To prove Theorem 2, we must first compute $\mathbb{E}[T_i(\pi)]$. This computation is addressed in Lemma 3.

Lemma 3 *Let $\pi = ((k_1, t_1), \ldots, (k_n, t_n))$ be a cyclic policy and let $T = T_{\mathrm{tr}} + \sum_{i=1}^{n} t_i$ be the period of the cyclic policy. Then*

$$\mathbb{E}[T_i(\pi)] = \frac{2}{\lambda_i} + \frac{T - t_i - t_i e^{-\lambda_i t_i}}{1 - e^{-\lambda_i t_i}}. \qquad (9)$$

PROOF. To compute $\mathbb{E}[T_i(\pi)]$, without loss of generality, fix an observation window at station i and call it observation window 0, or o_0. We further assume that o_0 contains the arrival of at least one event. We look at all observation gaps on the right of o_0. Any observation gap g_j contains the following parts, from left to right: 1. t_j^{left}, the overlap of g_j with the observation window on g_j's left end, 2. $T - t_i$, the first observation break (an observation break for station i is the time window between two consecutive visits to station i), 3. $0 \leq m < \infty$ additional

policy cycles (of length T each), and 4. t_j^{right}, the overlap of g_j with the observation window on g_j's right end. As an example, in Figure 2, the start and end of the observation gap g_j are marked with the two red lines, respectively. The parts t_j^{left}, the first observation break $T - t_i$, and t_j^{right} are also as marked. The gap g_j further contains two additional policy cycles, *i.e.*, $m = 2$. In this case, we say that g_j *spans* $m + 1 = 3$ policy cycles.
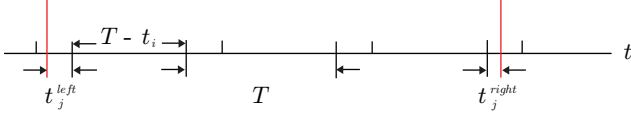

Fig. 2. Illustration of the components of an observation gap.

To compute $\mathbb{E}[T_i(\pi)]$, we split it into two steps: 1. compute the probability p_m of a gap g_j spanning $m + 1$ policy cycles for any $m \geq 0$, and 2. compute $\mathbb{E}[T_i(\pi)]$ as

$$\mathbb{E}[T_i(\pi)] = \sum_{m=0}^{\infty} \mathbb{E}_m p_m, \qquad (10)$$

in which $\mathbb{E}_m$ is the expected length of a gap g_j spanning $m + 1$ policy cycles. Note that Equation (10) holds as long as the expectations $\mathbb{E}[T_i(\pi)]$ and $\mathbb{E}_m$ are computed with the same underlying distribution. We can compute $\mathbb{E}_m$ with

$$\mathbb{E}_m = \mathbb{E}[t_j^{left}] + \mathbb{E}[t_j^{right}] + T - t_i + mT 2\mathbb{E}[t_j^{left}] + T - t_i + mT.$$

The second equality holds because $\mathbb{E}[t_j^{left}] = \mathbb{E}[t_j^{right}]$ by symmetry (*i.e.*, a time reversed Poisson process is again a Poisson process with the same arrival rate). To compute p_m, note that we never need to consider the left side of a gap g_j. This is true because as we look at an infinite sequence of consecutive gaps $g_1, \ldots, g_j, \ldots$, by assumption the left most observation window (which is o_0) overlapping with g_1 is already fixed. Once the right most observation window overlapping with g_1 is set (with certain probability), this explicitly fixes the left most observation window overlapping with g_2 and recursively, the left most observation window overlapping g_j. Therefore, the probability of g_j spanning $m + 1$ policy cycles is

$$p_m = e^{-m\lambda_i t_i}(1 - e^{-\lambda_i t_i}).$$

The first term in the expression for p_m, $e^{-m\lambda_i t_i}$, is the probability that g_j does not stop at $0, 1, \ldots, m-1$ policy cycles, where the probability of no event happening in each additional cycle in the sequence is $e^{-\lambda_i t_i}$. They can be combined due to the memoryless property of the exponential distribution. The second term $(1 - e^{-\lambda_i t_i})$ is the probability that at least one event happens in the right most observation window overlapping g_j. Noting that the terms $2\mathbb{E}[t_j^{left}] + T - t_i$ appear in all $\mathbb{E}_m$'s, we can rewrite $\mathbb{E}[T_i(\pi)]$ as

$$\mathbb{E}[T_i(\pi)] = 2\mathbb{E}[t_j^{left}] + T - t_i + \sum_{m=1}^{\infty} mT e^{-m\lambda_i t_i}(1 - e^{-\lambda_i t_i})$$

$$(11)$$

in which

$$\sum_{m=0}^{\infty} mT e^{-m\lambda_i t_i}(1 - e^{-\lambda_i t_i}) = T(1 - e^{-\lambda_i t_i})\sum_{m=1}^{\infty}\sum_{k=m}^{\infty} e^{-k\lambda_i t_i}$$

$$= T(1 - e^{-\lambda_i t_i})\sum_{m=1}^{\infty}\frac{e^{-m\lambda_i t_i}}{1 - e^{-\lambda_i t_i}} = \frac{T e^{-\lambda_i t_i}}{1 - e^{-\lambda_i t_i}}.$$

$$(12)$$

The computation of $\mathbb{E}[t_j^{left}]$ is carried out as follows. By assumption, at least one event happens during the given observation window of length t_i. Let the number of events within this t_i time be n (the probability of which is $Pr(n, \lambda_i t_i) = (\lambda_i t_i)^n e^{-\lambda_i t_i}/n!$) and let τ_1 be the arrival time of the first event among these n events. For each $n \geq 1$, the distribution of the n events is a uniform distribution in $[0, t_i]$. We have $Pr(\tau_1 > t)_{0 \leq t \leq t_i} = ((t_i - t)/(t_i))^n$, which gives us the pdf

$$p(\tau_1 = t)_{0 \leq t \leq t_i} = \frac{n(t_i - t)^{n-1}}{t_i^n}. \qquad (13)$$

Equation (13) gives us $\mathbb{E}[\tau_1] = t_i/(n+1)$. Then

$$\mathbb{E}[t_j^{left}] = \frac{\sum_{k=1}^{\infty}\frac{t_i}{k+1}Pr(k, \lambda_i t_i)}{1 - Pr(0, \lambda_i t_i)} = \frac{1}{1 - e^{-\lambda_i t_i}}\sum_{k=1}^{\infty}\frac{t_i(\lambda_i t_i)^k e^{-\lambda_i t_i}}{(k+1)!}$$

$$= \frac{1}{\lambda_i(1 - e^{-\lambda_i t_i})}(1 - e^{-\lambda_i t_i} - \lambda_i t_i e^{-\lambda_i t_i}) = \frac{1}{\lambda_i} - \frac{t_i e^{-\lambda_i t_i}}{1 - e^{-\lambda_i t_i}}.$$

$$(14)$$

Finally, plugging Equations (12) and (14) into Equation (11) yields Equation (9). $\qquad \square$

PROOF OF THEOREM 2. We now prove the quasi-convexity of of $\mathbb{E}[T_i(\pi)]$. For notational convenience, define $\gamma_i := \sigma/\lambda_i$. Note that we implicitly use the fact that all functions used in the proof are continuous. Substituting $T_{obs} = T - T_{tr}$ and $t_i = \gamma_i T_{obs}$ into the RHS of Equation (9) yields

$$\mathbb{E}[T_i(\pi)] = \frac{2}{\lambda_i} + \frac{T - t_i - (T - t_i)e^{-\lambda_i t_i} + (T - 2t_i)e^{-\lambda_i t_i}}{1 - e^{-\lambda_i t_i}}$$

$$= \frac{2}{\lambda_i} + \frac{T - t_i - t_i e^{-\lambda_i t_i}}{1 - e^{-\lambda_i t_i}}$$

$$= \frac{2}{\lambda_i} + \frac{T_{obs} + T_{tr} - \gamma_i T_{obs} - \gamma_i T_{obs}e^{-\lambda_i \gamma_i T_{obs}}}{1 - e^{-\lambda_i \gamma_i T_{obs}}}.$$

$$(15)$$

Noting that by scaling the unit of time, we may assume that $\lambda_i = 1$. Using this and letting $x := \gamma_i T_{obs}$ gives us

$$\mathbb{E}[T_i(\pi)] = 2 + \frac{T_{tr} + (\frac{1}{\gamma_i} - 1)x - xe^{-x}}{1 - e^{-x}} = 2 + \frac{T_{tr} + (\frac{1}{\gamma_i} - 2)x}{1 - e^{-x}} + x,$$

$$(16)$$

in which $T_{tr} > 0$ and $\gamma_i \in (0, 1)$. For convenience, we let $\alpha := T_{tr}$ and $\beta = 1/\gamma_i - 2$. Showing that $\mathbb{E}[T_i(\pi)]$ is quasi-convex is equivalent to showing that

$$f(x) := x + (\alpha + \beta x)/(1 - e^{-x})$$

is quasi-convex for $x > 0$,[2] $\alpha > 0$, and $\beta > -1$, the second

---

[2] In the rest of the proof, the domain of x is assumed to be $(0, \infty)$.

derivative of which is

$$f''(x) = \frac{e^x(\alpha(e^x+1)+\beta(e^x(x-2)+x+2))}{(-1+e^x)^3}. \tag{17}$$

Since $e^x(x-2)+x+2$ is strictly positive,[3] $f''(x) > 0$ for $\beta \geq 0$. Therefore, $f(x)$ is convex for $\beta \geq 0$. We are left to show that $f(x)$ is quasi-convex for $\beta \in (-1,0)$. We proceed by first establishing some properties of the function

$$g(x) = \alpha(e^x+1)+\beta(e^x(x-2)+x+2) \tag{18}$$

for $\alpha > 0$, and $\beta \in (-1,0)$. We have $g(x) \in C^\infty$ for $x \geq 0$, $g(0) = 2\alpha > 0$, $\lim_{x\to\infty} g(x) = -\infty$,

$$g'(x) = (\alpha+\beta x-\beta)e^x+\beta$$

and

$$g''(x) = (\alpha+\beta x)e^x.$$

Because $(\alpha+\beta x)$ is linear, monotonically decreasing and crosses zero at most once, and e^x is positive and strictly increasing, $g''(x)$ has at most a single local extrema (a maxima) before it crosses zero. Therefore, $g'(x)$ has at most two zeros and must first increase monotonically and then decrease monotonically, implying that $g(x)$ has at most three zeros. Since $g(0) > 0$ and $\lim_{x\to\infty} g(x) = -\infty < 0$, $g(x)$ has either one or three (but not two) zeros. For $g(x)$ to have three zeros, $g'(x)$ must have two zeros. Since $\lim_{x\to\infty} g'(x) = -\infty$ (because $\beta x e^x$ eventually dominates and $\beta < 0$), we must have $g'(0) < 0$. This is not possible because $g'(0) = \alpha > 0$. Therefore, $g'(x)$ can cross zero and change sign at most once, implying that $g(x)$ has a single zero. That is, $g(x)$ is positive for small x and then remains negative after crossing zero. Because $f''(x) = (e^x g(x))/(-1+e^x)^3$ and $e^x/(-1+e^x)^3$ is strictly positive, $f''(x)$ behaves similarly as $g(x)$ (i.e., $f''(0) > 0$, crosses zero only once as x increases, and stays negative after that). This implies that for every fixed $\alpha > 0$ and $\beta \in (-1,0)$, there exists $x_0 > 0$ such that $f(x)$ is convex on $x \in (0,x_0)$ and concave on $x \in (x_0,\infty)$. Now because $f(x) \to \infty$ for both $x \to 0^+$ and $x \to \infty$, and $f(1) < \infty$, $f(x)$ must have a single local minima (and therefore, a single global minima on $\mathbb{R}^+$). To see that this is the case, as $f(x)$ turns from convex to concave at $x = x_0$, we must have $f'(x_0) \geq 0$ because otherwise $f'(x) < 0$ for $x > x_0$ due to $f(x)$'s concavity. We then have $\lim_{x\to\infty} f(x) < \infty$, a contradiction. Thus, $f(x)$ has a single minimum on $x \in (0,x_0)$. Finally, to see that $f(x)$ is quasi-convex, we note that $\lim_{x\to\infty} f'(x) = 1 + \beta > 0$, implying that $f'(x) > 0$ on all $x \in (x_0,\infty)$. We then have that $f(x)$ is monotonically increasing on $x \in (x_0,\infty)$. From here, the quasi-convexity of $f(x)$ can be easily established following definitions. □

Interestingly, one can further show $\mathbb{E}[T_i(\pi)]$'s monotonic dependency with respect to λ_i, holding other parameters fixed.

Proposition 4 *For fixed* σ, *policy period* T, *and policy* π *given by Equation* (4), $\mathbb{E}[T_i(\pi)]$ *increases monotonically as* λ_i *increases.*

PROOF. Plugging $T_{\text{obs}} := T - T_{\text{tr}}$ and $\sigma := 1/(\sum_{i=1}^n (1/\lambda_i))$ into Equation (9) and treating it as a function of λ_i with T, T_{tr}, and σ all fixed, we get

$$f_N(\lambda_i) = \frac{2}{\lambda_i} + \frac{T - \frac{\sigma T_{\text{obs}}}{\lambda_i}(1+e^{-\sigma T_{\text{obs}}})}{1-e^{-\sigma T_{\text{obs}}}}, \tag{19}$$

the derivative of which is

$$f_N'(\lambda_i) = \frac{\sigma T_{\text{obs}} e^{-\sigma T_{\text{obs}}} + \sigma T_{\text{obs}} + 2e^{-\sigma T_{\text{obs}}} - 2}{\lambda_i^2(1-e^{-\sigma T_{\text{obs}}})}, \tag{20}$$

which is strictly positive for all positive σT_{obs} and arbitrary positive λ_i, implying that $f_N(\lambda_i)$ increases monotonically with respect to λ_i. □

Proposition 4 implies that Equation (2) is always determined by the Poisson process with the largest intensity. Therefore, we only need to look at the single largest λ_i when we optimize the second objective.

IV. COMPUTATIONAL EXPERIMENTS

Our experimental setup is a direct adaptation of the UAV monitoring application illustrated in Figure 1. The UAV is tasked to fly continuously along the six locations of interest and hover over each location for some time to capture stochastically occurring events at these locations. The input consists of historical data for the event arrival rates λ_i's at the event locations and the time needed for traveling between the event locations. Table II lists λ_i's and $\tau_{i,j}$'s (the ground truth) for the experiment. The time unit is hour (hr). Figure 3 illustrates the stochastic nature of the event arrivals (note that these are standard simulations of the exponential distributions and Poisson processes). In addition to the large range of average arrival rates at different stations (e.g., events arrive at station 3 five times more frequent than they do at station 1), the stochastic arrival times can vary greatly within the same station. The UAV must balance the amount of data collected at all stations despite the different arrival rates while not incurring large delays in event observations between consecutive visits to the same location.

The two goals of our experimental effort are to *(i)* further demonstrate the correctness of our theoretical developments, and *(ii)* investigate the performance of the proposed algorithm with respect to various measures. For the first goal, we measure how well our theoretical predictions hold up by comparing simulation outcome to our analytical result side by side. For the second, we verify that the computed optimal policy achieves all the design expectations. We also compare it with a non-optimal policy and contrast their performances.

The source code for our simulation software was developed using the Java programming language, and the simulation software itself was executed on a computer with a 1.3GHz Intel Core i5 CPU and 4GB memory. Mathematica 9 was used

---

[3]To see this, let $h(x) = e^x(x-2)+x+2$; then $h(0) = 0$, $h'(0) = 0$, and $h''(x) = xe^x > 0$ for all $x > 0$. Therefore, $h'(x) > 0$ and $h(x) > 0$ for all $x > 0$.
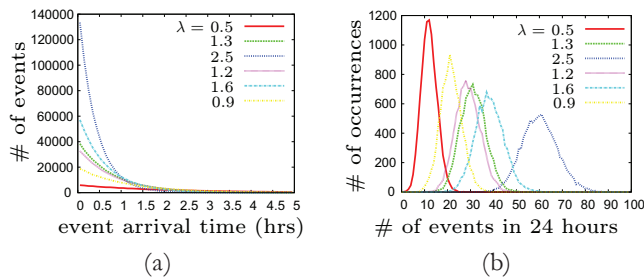
Fig. 3. (a) Histogram over the event arrival times since the last event arrival for the Poisson processes in our experiment over a time horizon of 10000 days. The bucket size (on the x axis) is 0.1 hour. (b) Histogram over the number of events arriving in an 24-hour window for the different Poisson processes over 10000 runs.

for computing the optimal policy using the gradient descent optimization procedure.

TABLE II
THE GROUND TRUTH (EVENT ARRIVAL RATES AND TRAVEL TIMES) USED IN OUR SIMULATIONS.

	Station					
	1	2	3	4	5	6
λ_i (1/hr)	0.5	1.3	2.5	1.2	1.6	0.9
$\tau_{i,i+1 \mod 6}$ (hrs)	0.15	0.25	0.1	0.3	0.2	0.2

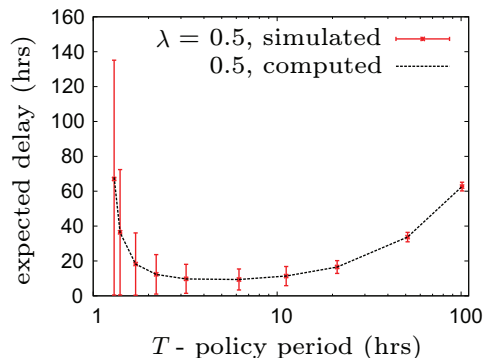## A. Empirical Verification of Theorem 2

Fig. 4. The simulated versus computed values for $\mathbb{E}[T_i(\pi)]$. We observe that the mean of the simulated runs agrees very well with the value computed directly from Equation (9) for all choices of T's, wheres the variance grows larger as $T \to T_{tr}$.

In addition to the proof, we empirically check the correctness of Theorem 2 via simulations (simulation validation of Lemma 1 is omitted given its obvious correctness). Our first computational experiment validates Equation (9) by performing both simulation and direct computation side by side and comparing the results, for the aforementioned case. In simulation, for each fixed $T \in \{1.3, 1.4, 1.7, 2.2, 3.2, 6.2, 11.2, 21.2, 51.2, 101.2\}$, we simulated the Poisson process for enough number of periods

(roughly 2×10^5 in the worst case) to gather at 2000 delays by simulating the the policy. This gave us 2000 samples of the random variable $T_i(\pi)$ from which we computed the mean and standard deviation. Direct computation based on Equation (9) were also carried out. To avoid cluttering the presentation, only $\lambda = 0.5$ was used (plots for other λ are similar).

The results of this simulation study are presented in Figure 4, in comparison with the optimal policy that is directly computed using the gradient descent procedure. Notice that the expected delay in simulation results match that of the computed policy exactly for all choices of T's. We also observe from the simulation study that the variance of the delay increases as T approaches T_{tr}. This should be intuitively clear, since, as $T_{obs} = T - T_{tr} \to 0^+$, the length of each observation window decreases when compared to T_{tr}; in fact, the ratio of the two approaches zero, which leads to the unbounded increase in the variance of the number of events observed in a given observation window.
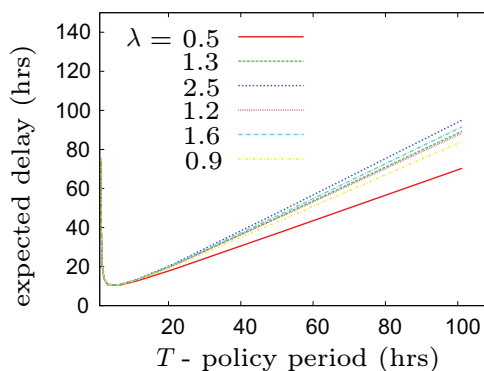

Fig. 5. The computed $\mathbb{E}[T_i(\pi)]$ for $\lambda_1, \ldots, \lambda_6$ and $T \in [1.3, 101.2]$.
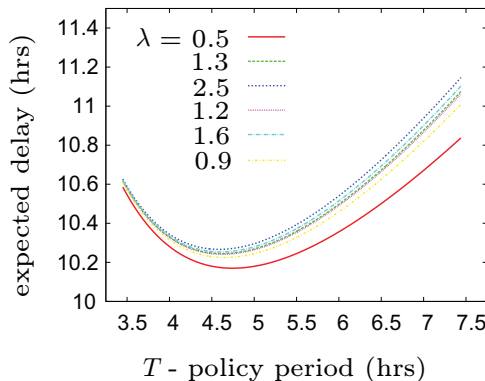

Fig. 6. The computed $\mathbb{E}[T_i(\pi)]$ for $\lambda_1, \ldots, \lambda_6$ and $T \in [3.25, 7.75]$ with $\Delta T = 0.025$ increments.

After empirically verifying that Equation (9) is accurate, we shift our attention to the quasi-convexity of Equation (9) and its monotonicity in λ_i. We compute $\mathbb{E}[T_i(\pi)]$ for all six λ_i's and plot the result at two different scales in Figure 5 and 6. Figure 5 shows that $\mathbb{E}[T_i(\pi)]$ is quasi-convex for all λ_i's. Figure 6, the zoomed-in version of Figure 5, further reveals

that $\mathbb{E}[T_i(\pi)]$ depends on λ_i monotonically for fixed period T, confirming the claim of Proposition 4.

B. The Performance of the Proposed Algorithm

To compute the optimal cyclic patrolling policy's parameters, by Proposition 4 we only need to look at $\mathbb{E}[T_i(\pi)]$ for $\lambda_i = 2.5$. The period T that minimizes Equation (9) for $\lambda_i = 2.5$ can be easily computed using standard gradient descent methods. Our computation yields $T^* = 4.59$. The corresponding policy is then defined by $\pi = (1.18, 0.45, 0.24, 0.49, 0.37, 0.67)$. Since our theoretical results guarantee the performance of the

TABLE III
COMPARISON OF AN OPTIMAL POLICY WITH A NON-OPTIMAL ONE.

Station	$\alpha_i(\cdot)$		$\mathbb{E}[T_i(\cdot)]$ (hrs)		$\sigma_{\mathbb{E}[T_i(\cdot)]}$ (hrs)	
	π	π'	π	π'	π	π'
1	0.17	0.06	10.1	18.3	7.5	16.0
2	0.17	0.16	10.1	8.9	7.4	6.4
3	0.16	0.31	10.3	5.9	7.7	2.9
4	0.16	0.15	10.2	9.3	7.9	6.5
5	0.17	0.20	10.2	7.6	7.6	5.0
6	0.17	0.11	10.1	11.3	7.5	8.7

patrolling policy, we carried out a single simulation experiment in comparing the optimal policy with non-optimal policies. For our comparison, we evenly distributed $T_{\mathrm{obs}} = T - T_{\mathrm{tr}} = 3.39$ among the stations and obtained an alternative policy π' that spends 0.57 (hours) at each station per cycle. We simulated both policy for 100000 policy periods. The simulation results $(\alpha(\cdot), \mathbb{E}[T_i(\cdot)]$, and the standard deviation of $\mathbb{E}[T_i(\cdot)]$, denoted $\sigma_{\mathbb{E}[T_i(\cdot)]}$) are listed in Table III. The result speaks for itself: Under the optimal policy π, $\alpha_i(\pi)$'s are uniform across all stations. At the same time, $\mathbb{E}[T_i(\pi)]$'s are also very uniform and are all about twice of the policy cycle time $T^* = 4.59$. On the other hand, under policy π', station 1 often gets neglected with an $\alpha_1(\pi') = 0.06$ and a $\mathbb{E}[T_1(\pi')] = 18.3$, which are both much worse than those for the optimal policy π.

V. Conclusions

In this paper, we introduced a novel persistent monitoring problem and data collection in which the arrivals events at multiple stations are driven by stochastic processes. We studied the performance of cyclic policies on two objectives: *(i)* balancing the average number of events to be collected at each station so that no station receives insufficient or excessive monitoring effort, and *(ii)* minimizing the maximum delay in observing two consecutive events generated by the same process between policy cycles. We focused on an important special case where the locations to be visited form a closed chain. We showed that such a problem admits a unique cyclic policy that optimizes both objectives. Moreover, we established that the the second and more complex objective turned out to be quasi-convex, allowing efficient computation of the optimal policy with standard gradient descent methods. We conjecture that these results can be applied to the general

case where the locations are connected in an arbitrary way, rather than the closed chain configuration. We conjecture that in this general case, the optimal solution can be obtained by first solving a Traveling Salesman Problem (TSP), and then computing a schedule along the optimal TSP tour using the algorithm we propose in this paper.

References

[1] N. Michael, E. Stump, and K. Mohta, "Persistent surveillance with a team of mavs," in *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 2708–2714.

[2] R. N. Smith, M. Schwager, S. L. Smith, B. H. Jones, D. Rus, and G. S. Sukhatme, "Persistent ocean monitoring with underwater gliders: Adapting sampling resolution," *Journal of Field Robotics*, vol. 28, no. 5, pp. 714–741, September-October 2011.

[3] 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*, 2012, to appear.

[4] E. Arvelo, E. Kim, and N. C. Martins, "Memoryless control design for persistent surveillance under safety constraints," September 2012, available at http://arxiv.org/abs/1209.5805.

[5] C. G. Cassandras, X. Lin, and X. Ding, "An optimal control approach to the multi-agent persistent monitoring problem," *IEEE Transactions on Automatic Control*, vol. 58, no. 4, pp. 947–961, April 2013.

[6] A. Girard, A. Howell, and J. Hedrick, "Border patrol and surveillance missions using multiple unmanned air vehicles," in *Proc. 43rd IEEE Conference on Decision and Control*, 2004, pp. 620–625.

[7] B. Grocholsky, J. Keller, V. Kumar, and G. Pappas, "Cooperative air and ground surveillance," *IEEE Robotics and Automation Magazine*, vol. 13, no. 3, pp. 16–25, Sep 2006.

[8] X. Lan and M. Schwager, "Planning periodic persistent monitoring trajectories for sensing robots in gaussian random fields," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA 13)*, May 2013, pp. 2407–2412.

[9] N. Nigam and I. Kroo, "Persistent surveillance using multiple unmanned air vehicles," in *Proc. IEEE Aerospace Conference*, 2008, pp. 1–14.

[10] S. L. Smith, M. Schwager, and D. Rus, "Persistent robotic tasks: Monitoring and sweeping in changing environments," *IEEE Transactions on Robotics*, vol. 28, no. 2, pp. 410–426, April 2012.

[11] D. E. Soltero, M. Schwager, and D. Rus, "Generating informative paths for persistent sensing in unknown environments," in *Proc. of the International Conference on Intelligent Robots and Systems (IROS 12)*, October 2012, pp. 2172–2179.

[12] H. Choset, "Coverage of known spaces: The boustrophedon cellular decomposition," *Autonomous Robots*, vol. 9, pp. 247–253, 2000.

[13] ——, "Coverage for robotics - a survey of recent results," *Annals of Mathematics and Artificial Intelligence*, vol. 31, pp. 113 – 126, 2001.

[14] Y. Gabriely and E. Rimon, "Competitive on-line coverage of grid environments by a mobile robot," *Computational Geometry*, vol. 24, no. 3, pp. 197–224, 2003.

[15] W.-P. Chin and S. Ntafos, "Optimum watchman routes," *Information Processing Letters*, vol. 28, pp. 39–44, 1988.

[16] P. Hokayem, D. Stipanovic, and M. Spong, "On persistent coverage control," in *Proc. 46th IEEE Conference on Decision and Control*, 2008, pp. 6130–6135.

[17] S. Ntafos, "Watchman routes under limited visibility," *Computational Geometry*, vol. 1, pp. 149–170, 1991.

[18] J. A. Fuemmeler and V. V. Veeravalli, "Smart sleeping policies for energy-efficient tracking in sensor networks," *Networked Sensing Information and Control*, 2008.

[19] Y. He and E. K. P. Chong, "Sensor scheduling for target tracking in sensor networks," in *Proc. 43rd IEEE Conference on Decision and Control*, 2004, pp. 743–748.

[20] A. O. H. III, C. M. Kreucher, and D. Blatt, "Information theoretic approaches to sensor management," *Foundations and Applications of Sensor Management*, 2008.

[21] J. L. Ny, M. A. Dahleh, E. Feron, and E. Frazzoli, "Continuous path planning for a data harvesting mobile server," in *47th IEEE Conference on Decision and Control*, 2008, pp. 1489–1494.