Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 52-20 (2019) 217–222

A sub-modular receding horizon approach to persistent monitoring for a group of mobile agents over an urban area

Navid Rezazadeh * Solmaz S. Kia *

* Department of Mechanical and Aerospace Engineering, University of California Irvine, Irvine, CA 92697, {nrezazad,solmaz}@uci.edu.

Abstract: We consider the problem of persistent monitoring of a finite number of inter-connected geographical nodes for event detection via a group of heterogeneous mobile agents. We assume that the probability of the events occurring at the geographical points of interest follow known Poisson processes. We tie a utility function to the probability of detecting an event in each point of interest and use it to incentivize the agents to visit the geographical nodes with higher probability of event occurrence. We show that the design of an optimal monitoring policy that specifies the sequence of the geographical nodes and time of visit of those nodes for each mobile agent so that the utility of event detection over a mission horizon is maximized is an NP-hard problem. To reduce the time complexity of constructing the feasible set of the optimal approach and also to induce robustness to changes in event occurrence and other operational factors, we consider a receding horizon approach. We note that, with the number of agents growing, the cost of finding the optimal path grows exponentially even with shortened horizon. To overcome this issue, we introduce a sub-modular optimization approach that has a polynomial time complexity and also comes with a known sub-optimality lower bound. We demonstrate our results through simulations.

Keywords: Multi-agent systems, Persistent monitoring, Submodular optimization.

1. INTRODUCTION

In extension of cities and technology there is always a need for surveillance to monitor for incidences of interest. Traditionally, the surveillance systems are stationary, and are only able to cover limited areas. To achieve reliable monitoring via stationary sensors in a large area, it is necessary to deploy a huge number of sensors. Even in cases where the cost is not a major prohibitive factor, with the current technology, the communication bandwidth utilization certainly is. Therefore, to solve the coverage within the limits of the system, use of mobile sensors, which the infrastructure can move within the urban area is of interest (especially aerial sensors that have wide measurement zones). The aim is to compensate for the lack of full spatial coverage at all times by context-aware temporal dynamic distribution of a set of mobile sensors. In this paper, we consider the problem of designing a dispatch policy for mobile agents that automatically orchestrates the topological distribution of the mobile sensors such that the 'best' service for the global monitoring task is provided within the constraints of the network.

We consider a monitoring scenario in which the areas of interest for monitoring via mobile agents are given as a set of known finite geographical nodes that are connected to each other with known and pre-specified corridors, see Fig. 1. We assume that the probability of the events occurring at the geographical points of interest follow known Poisson processes whose rate is extracted from historical data. This rate can also change based on data that becomes available. We recall that the Poisson process is a widely-used counting process for scenarios where we are counting the occurrences of certain events that happen at a certain rate, but completely at random (without a certain structure) [1]. To provide a context aware monitoring, we tie a utility function to the probability of detecting an event in each point of interest and use it to incentivize the agents to visit the geographical nodes with higher probability of event occurrence. We assume that the mobile agents are heterogeneous in a sense that their traveling time and also the time to detect an event are different. We show that the optimal monitoring policy that specifies the sequence of the geographical nodes and time of visit of those nodes for each mobile agent so that the utility of event detection over a mission horizon is maximized is an NP-hard problem. Our ultimate goal in this paper is to design a multi-agent persistent monitoring solution that detects maximum number of events over a given mission horizon with a polynomial cost.

Our work is related to the problem of persistent monitoring/patrolling via mobile agents. A multi-agent persistent monitoring is a scheme in which a set of mobile robots are dispatched in a designated area to gather information. The agents move among a set of defined points in the area and gather information via interaction with the points, which an interaction can range from capturing a photo of the point to communicating with a local agent in that point. In a persistent monitoring setup, the agents are anticipated to have some level of intelligence such that they can be able to maintain robustness in case of unwanted incidences. This enables them to continue fulfilling their tasks without requiring constant supervision. Persistent monitoring of geographical areas for event detection is of prime interest in many applications such as discovering forest fires [2] and oil spillage in their early stages [3], locating endangered animals in a large habitat [4]. Long term multi-agent patrolling of an area offers a low cost and effective monitoring solution for these applications. In a single agent patrolling of a set of connected nodes in an area, the complexity of finding the 'best' route is the same as the complexity of traveling sales man problem and grows exponentially with the number of the nodes [25]. The problem of optimal multi-agent patrolling inherently is more complex

⋆ This work is supported by NSF award IIS-SAS-1724331.

Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2019.12.161

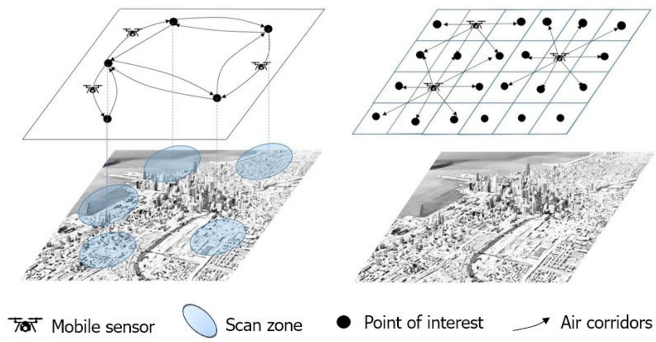


Fig. 1. Examples of a set of geographical points of interest and the edges between them. Finite number of points to monitor in a city can be restricted to some particular scanning zones (the picture on the left) or the cell partitioned map of the city (the picture on the right).

than single agent patrolling since each agent's patrolling scheme depends on other agents' policy. The problem of multi-agent patrolling was formalized in [5, 6]. Because finding an optimal long term patrolling scheme is not tractable, posing limitations on the structure of nodes and how agents can move between them makes it possible to find the optimal solution. Optimally of scheduling on line and cycle graphs were extensively studied in [7, 8]. In a context where the location of nodes are determined and the routes between then are undetermined, Rus et al. proposed an optimal cyclic patrolling scheme [9, 10]. While multi-agent patrolling was considered as a task scheduling problem, Cassandras et al. pushed it to continuous domain where the movement trajectory of the agents is subject of design [11, 12]. Traditionally the time that nodes were not visited by an agent has been the criterion to design the monitoring algorithms (minimizing nodes' idle time). [13] added the flavor of uncertain condition in the nodes where agents are subject to removal and added it to criterion on designing the patrolling scheme.

Instead of using the customary criterion of minimizing the idle time of the nodes, which comes with the underlying assumption that the rate of events happening at all the nodes is uniform, we design a dispatch policy that uses a Poisson random variable as the model for arrival of valuable information at the nodes. Hence, we base our total reward on Poisson distribution. We show that maximizing the proposed reward is an NP-hard problem. Next, we show that the reward function is a monotone submodular set function. Then, based on this result, we propose a receding horizon sequential greedy algorithm to compute a sub-optimal dispatch policy with a polynomial computation cost and guaranteed bound on optimality. The receding horizon nature of our solution induces robustness to uncertainties of the environment. In recent years, submodular optimization has been widely used in sensor and actuator placement problems [15, 16, 17, 18, 19, 20]. In comparison to the sensor/actuator placement, the challenge in our work is that the assigned policy per each mobile agent over the receding horizon is a vector rather than a point. To deal with this challenge, we use the so-called matroid constraint [21] approach to design our sub-optimal submodular-based policy. We demonstrate our results through two simulation studies. Due to space limitation, the proof of our results are not given in the paper and will appear elsewhere.

2. PRELIMINARIES

Notation: We let $\mathbb{R}$, $\mathbb{R}_{>0}$, $\mathbb{R}_{\geq 0}$, $\mathbb{Z}$ denote the set of real, positive real, non-negative real, integer, respectively. Given the sets $\mathcal{A}$ and $\mathcal{B}$, $\mathcal{B} \subset \mathcal{A}$ means that $\mathcal{B}$ is a subset of $\mathcal{A}$. Moreover, $2^{\mathcal{A}}$ is the set of all the subsets of $\mathcal{A}$. And finally, given an element $a \in \mathcal{A}$, for simplicity we represent $\mathcal{B} \cup \{a\}$ by $\mathcal{B} \cup a$. Given an event set $\mathcal{V}$ and $e \in \mathcal{V}$, $P(e) : \mathcal{V} \to [0,1]$ denotes the probability of event e happening. Let discrete random variable X take values in $\mathcal{X} = \{x_1, x_2, ...\}$, then the function $P(X = x_i) : \mathcal{X} \to [0,1]$ shows the probability of X taking value $x_i \in \mathcal{X}$ and expected value of X is defined as $E[X] = \sum_{x_i \in \mathcal{X}} P(X = x_i) x_i$. We denote a sequence of m increasing real numbers $(t_1, ..., t_m)$ (i.e., $t_k \leq t_{k+1}$ for $k \in \{1, ..., m\}$) by $(t)_1^m$. Given $(t)_1^n$ and $(v)_1^m$ we denote by $(t)_1^n \oplus (v)_1^m$ their concatenated increasing sequence, i.e., for $(u)_1^{n+m} = (t)_1^n \oplus (v)_1^m$ we have that any u_k, $k \in \{1, \cdots, n+m\}$ is either in $(t)_1^n$ or $(v)_1^m$ or is in both of $(t)_1^n$ and $(v)_1^m$. We assume that $(u)_1^{n+m}$ preserves the relative labeling of $(t)_1^n$ or $(v)_1^m$, i.e., if t_k and t_{k+1}, $k \in \{1, \cdots, n-1\}$ (resp. v_k and v_{k+1}, $k \in \{1, \cdots, m-1\}$) correspond to u_i and u_j in $(u)_1^{n+m}$, then $i < j$.

Poisson process: Here, we briefly review some properties of the Poisson random process following [1]. Counting process $\{C(t,t_0), t \geq t_0\}$ is a stochastic process that keeps a record of number of events happened during the time interval $(t_0, t]$. $C(t,t_0)$ is defined to be 0 with probability 1, which means that we are considering only events happened at strictly positive times. $\{C(t,t_0), t \geq t_0\}$ is non-negative, integer valued ($C(t,t_0) \in \mathbb{Z}$) and increasing. A counting process $\{C(t,t_0), t \geq t_0\}$ is said to be a Poisson process with a rate $\lambda \in \mathbb{R}_{>0}$, denoted by $C(t,t_0) \sim \texttt{Poisson}(\lambda(t - t_0))$, if it has an exponential distribution function

$$P(C(t,t_0) = n) = \frac{(\lambda(t-t_0))^n \mathrm{e}^{-\lambda(t-t_0)}}{n!}.$$

For a Poisson counting process, $C(t,t_0)$ has stationary and independent increments, and also $\texttt{E}[C(t,t_0)] = \lambda(t - t_0)$. The occurrence time (arrival time) of the i^{th} Poisson event is also a random variable, which we denote by $W_i(t_0) \in \mathbb{R}_{>0}$. Starting at t_0, the probability of no event taking place in time interval $(t_0, t]$ is given by

$$P(W_1(t_0) > t) = P(C(t,t_0) = 0) = \mathrm{e}^{-\lambda(t-t_0)}.$$

Therefore, the probability of at least one event taking place in time interval $(t_0, t]$ is given by

$$P(W_1(t_0) \leq t) = 1 - \mathrm{e}^{-\lambda(t-t_0)}.$$

In general, for any $i \in \mathbb{Z}_{>0}$ W_i follows a Gamma distribution $W_i \sim \texttt{Gamma}(\lambda, i)$. Let $C(t_v, t_0) = q$, i.e., the number of events occurred in $(t_0, t_v]$ is q. Then, due to stationary and independent increment of the Poisson process for any $t_v \in (t_0, t)$ we have

$$P(C(t,t_0) = n + q | C(t_v, t_0) = q) = P(C(t, t_v) = n), \tag{1a}$$

$$P(W_{q+1}(t_0) > t | C(t_v, t_0) = q) = P(W_1(t_v) > t) = \mathrm{e}^{-\lambda(t-t_v)}, \tag{1b}$$

$$P(W_{q+1}(t_0) \leq t | C(t_v, t_0) = q) = 1 - \mathrm{e}^{-\lambda(t-t_v)}. \tag{1c}$$

When the rate is time-varying, $\lambda : \mathbb{R}_{\geq 0} \to \mathbb{R}_{>0}$, the distribution of the counting process is given by $\texttt{Poisson}(\int_{t_0}^{t} \lambda(\tau) \mathrm{d}\tau)$.

Submodular function: Submodularity is a property of a set functions that shows diminishing reward as new members are being introduced to the system. Here, we provide a brief review of the submodular functions, following [22].

Definition 1. A set function $g : 2^{\mathcal{P}} \to \mathbb{R}$ is monotone decreasing if for all $\mathcal{P}_1, \mathcal{P}_2 \subset \mathcal{P}$ the following is satisfied

Algorithm 1 Sequential Greedy Algorithm

procedure $\mathsf{SGOpt}(\mathcal{P}, M)$
  Init: $\bar{\mathcal{P}} \leftarrow \emptyset,\ i \leftarrow 0$
  loop:
  while $i \leq M$ **do**
    $\underset{\mathsf{p} \subset \mathcal{P}}{\text{maximize}} \quad \Delta_g(p|\bar{\mathcal{P}})$.
    if $\Delta_g(\mathsf{p}|\bar{\mathcal{P}}) \leq 0$ **then**
      Break
    end if
    $\bar{\mathcal{P}} \leftarrow \bar{\mathcal{P}} \cup p$.
    for i=1:M **do**
      if $\mathsf{p} \in \mathcal{P}^i$ **then**
        $\mathcal{P} \leftarrow \mathcal{P} \backslash \mathcal{P}^i$
        Break
      end if
    end for
  end while
  Return $\bar{\mathcal{P}}$.
end procedure

$$\mathcal{P}_1 \subset \mathcal{P}_2 \quad \text{if and only if} \quad g(\mathcal{P}_1) \geq g(\mathcal{P}_2).$$

A set function $\Delta_g : (\bar{\mathcal{P}}, \mathsf{p}) \to \mathbb{R}$, $\Delta_g(\mathsf{p}|\bar{\mathcal{P}}) = g(\bar{\mathcal{P}} \cup \mathsf{p}) - g(\bar{\mathcal{P}})$ for $\forall \bar{\mathcal{P}} \in 2^{\mathcal{P}}$ and $\forall \mathsf{p} \in \mathcal{P}$, shows the increase in value of the submodular set function g going from set $\bar{\mathcal{P}}$ to $\bar{\mathcal{P}} \cup \mathsf{p}$.

Theorem 1. (See [22]). A set function $g : 2^{\mathcal{P}} \to \mathbb{R}$ is submodular if and only if for two sets $\mathcal{P}_1, \mathcal{P}_2$ satisfying $\mathcal{P}_1 \subset \mathcal{P}_2 \subset \mathcal{P}$, and for $p \notin \mathcal{P}_2$ we have

$$\Delta_g(p|\bar{\mathcal{P}}_1) \geq \Delta_g(p|\bar{\mathcal{P}}_2). \tag{2}$$

□

It is well-know that the optimization problem

$$\max_{\bar{\mathcal{P}} \subset \mathcal{P}} g(\bar{\mathcal{P}}), \quad s.t., \tag{3a}$$

$$|\bar{\mathcal{P}} \cap \mathcal{P}^i| \leq 1, \quad i \in \{1, \ldots, M\}, \tag{3b}$$

where $\mathcal{P}^1, \cdots, \mathcal{P}^M$ are collection of K disjoint sets that satisfy $\bigcup_{i=1}^{M} \mathcal{P}^i = \mathcal{P}$, is NP-hard [23]. When the objective function g in (3) is submadular, however, it is shown in the literature that a sequential greedy member selection algorithm (see Algorithm 1) results in a sub-optimal solution with a guaranteed bound on optimality.

Theorem 2. (See [22]). Let g in the optimization problem (3) be a monotone submodular set function. Suppose $g(\mathcal{P}^\star)$ is the global maximum of (3). Let $\bar{\mathcal{P}}$ be the output of Algorithm 1.Then, $g(\bar{\mathcal{P}}) \geq \frac{1}{2} g(\mathcal{P}^\star)$. □

3. PROBLEM DEFINITION

3.1 Problem setting

We consider a group of M mobile sensors (each referred henceforth as mobile agent or simply agent) are deployed to monitor N geographical points (each referred henceforth as geographical node or simply node) on a $\mathbb{R}^2$ or $\mathbb{R}^3$ space. We assume that every agent and every geographical node has a unique identification belonging to, respectively, the sets $\mathcal{A} = \{1, \cdots, M\}$ and $\mathcal{V} = \{1, \cdots, N\}$. At each geographical node $i \in \mathcal{V}$ there is an event that takes place in random, whose probability of occurrence follows a Poisson random process $C_i(t, t_{v,i}) \sim \mathtt{Poisson}(\lambda_i(t - t_{v,i}))$ where $t_{v,i} \in \mathbb{R}_{\geq 0}$ is the last time node i is visited by a mobile agent $j \in \mathcal{A}$. The rate of event occurrence $\lambda_i \in \mathbb{R}_{>0}$ can be specified based off of the historical data. This rate can change based on data that has become available from other sources. The rate also can be manipulated to steer the agents in certain direction on the map. Finally, $\lambda_i \in \mathbb{R}_{>0}$ can be time varying. To simplify the exposition, we demonstrate our results for fix occurrence rate. The results generalizes to time-varying case by using $C_i(t, t_{v,i}) \sim \mathtt{Poisson}(\int_{t_{v,i}}^{t} \lambda_i(\tau) \mathrm{d}\tau)$.

We assume that it takes $\delta^j \in \mathbb{R}_{>0}$ time for each mobile agent $j \in \mathcal{A}$ to process its sensor measurement collected upon its arrival time at a geographical node. For example, δ^j can be the time a mobile agent needs to process and detect an event from an image shot upon its arrival time. To incentivize the agents to visit and scan a geographical node $i \in \mathcal{V}$, we associate the reward function

$$R_i(t) = \begin{cases} 0, & t = t_{s,i}, \\ \psi_i(t - t_{s,i}), & t > t_{s,i}, \end{cases} \tag{4}$$

where

$$\begin{aligned} \psi_i(t - t_{s,i}) =& P(W_{q_i+1} \leq t \big| C_i(t_{s,i}, t_0) = q_i) \\ =& 1 - \mathrm{e}^{-\lambda_i(t - t_{s,i})}. \end{aligned} \tag{5}$$

Here $t_{s,i}$ is the latest scan time of node i and q_i is the number of the events that has been observed at geographical node i in the interval $(t_0, t_{i,s}]$. We note that $R_i(t)$ resets to 0 after an agent arrives and scans the node, and monotonically increases at an exponential rate of λ_i afterward. Moreover, we note here that $\psi_i(t - t_{s,i})$ is a measure of finding at least one event in node i. Upon arrival of any agent $j \in \mathcal{A}$ at time $t_v \in \mathbb{R}_{\geq 0}$ at node $i \in \mathcal{V}$, the agent immediately scans for the events (e.g., takes a picture) and the reward $R_i(t_v)$ is scored for the patrolling team $\mathcal{A}$ and $t_{s,i}$ is set to t_v.

Assumption 1. If more than one agent arrive at node $i \in \mathcal{V}$ and scan it at the same time t_v, the reward collected for the team is still $R_i(t_v)$ (note that after the first scan $t_{s,i}$ sets to t_v). Furthermore, every agent $j \in \mathcal{A}$ should spend δ^j amounts of time at the node to complete its measurement processing and event detection task. During the processing time the agent cannot scan for events.

The geographical nodes are connected via a set of pre-specified known corridors (each referred henceforth as an edge), and the mobile agents are confined to travel through these edges in order to traverse from one node to another (see Fig. 1). For example, in a smart city setting regulations can restrict the admissible routes between the geographical nodes. Also, depending on the vehicle type, agents may have to take different paths going from one node to another. We let $\mathcal{E}_{i,j}$ be the set of edges between nodes $i, j \in \mathcal{V}$. We assume that each geographical node is connected at least through one edge to another geographical node. We also add a self-loop to each node $i \in \mathcal{V}$, i.e., $|\mathcal{E}_{i,i}| = 1$. The self-loops are introduced to allow our motion planning policy to let agents to stay put and continue scanning at a node. For every node $i \in \mathcal{V}$, we let $\mathcal{N}_i$ be the set of its neighboring nodes that are connected to it via an edge. This neighbor set includes the node i itself, as well. We assume that the travel time $\tau_{i,j}^k(l) \in \mathbb{R}_{>0}$ between every node $i, j \in \mathcal{V}$ for agent $l \in \mathcal{A}$ along any edge $k \in \mathcal{E}_{i,j}$ is known, and $\tau_{i,i} = 0$ (travel time along a self-loop edge is zero). We assume that the set of mobile robots is heterogeneous, therefor the travel time differs for different agents on the same edge. The travel time at each edge also can change during the mission time. We assume that the travel time along every edge and for each agent is proportional to the length of the edge, and time to go from a point along the edge to its end nodes is also known.

3.2 Objective statement

The ultimate goal in the persistent monitoring problem over the setting described in previous subsection is to detect maximum number of events over a given mission horizon. By definition of the reward function (4), this objective equivalently can be expressed as designing a patrolling policy (what sequence of nodes to visit at what times) for the group of mobile agents $\mathcal{A}$ so that the group collects maximum possible reward over the mission horizon. In all the policy designs (optimal and sub-optimal) we consider the following assumption, as well.

Assumption 2. Each agent $i \in \mathcal{A}$ can only move to neighboring nodes of a previously visited node. Moreover, every agent scans the node that it visits.

The optimal monitoring policy, under Assumptions 1 and 2, over a given mission time should assign a sequence of N_f^i nodes to each agent $i \in \mathcal{A}$. Let $\mathsf{n}^i = \left[\mathsf{n}^i(0), \mathsf{n}^i(1), \cdots, \mathsf{n}^i(N_f^i)\right]^\top$ be the sequence of the nodes visited by agent $i \in \mathcal{A}$, with $\mathsf{n}^i(0)$ being the first node that agent i visits (starts from). Because of the Assumption 2, we have $\mathsf{n}^i(j+1) \in \mathcal{N}_{\mathsf{n}^i(j)}$, for all $j \in \{1, 2, ..., N_f^i - 1\}$. Let $\mathbf{T}^i = \left[T^i(0), T^i(1), \cdots, T^i(N_f^i)\right]$ be the visiting time associated with visiting sequence n^i of agent $i \in \mathcal{A}$. Given the starting location, let $\mathcal{P}^i$ be the set of all the feasible tuples $\mathsf{p}^i = (\mathsf{n}^i, \mathbf{T}^i, i)$ over the mission horizon for agent $i \in \mathcal{A}$ and let $\mathcal{P} = \bigcup_{i=1}^M \mathcal{P}^i$. Then, for any $\bar{\mathcal{P}} \subset \mathcal{P}$, the collected reward $\mathsf{R} : 2^{\mathcal{P}} \to \mathbb{R}_{>0}$ is

$$\mathsf{R}(\bar{\mathcal{P}}) = \sum_{\forall \mathsf{p} \in \bar{\mathcal{P}}} \sum_{j=1}^{|\mathsf{n}_\mathsf{p}|} R_{\mathsf{n}_\mathsf{p}(j)}(T_\mathsf{p}(j)), \tag{6}$$

with $\mathsf{p} = (\mathsf{n}_\mathsf{p}, \mathbf{T}_\mathsf{p}, i_\mathsf{p})$. Then, given (6), the optimal policy to maximize the team collected reward over a given mission horizon is given by

$$\mathcal{P}^\star = \underset{\bar{\mathcal{P}} \subset \mathcal{P}}{\operatorname{argmax}}\, \mathsf{R}(\bar{\mathcal{P}}), \quad \text{s.t.} \tag{7a}$$

$$|\bar{\mathcal{P}} \cap \mathcal{P}^i| \leq 1, \quad i \in \{1, \dots, M\}. \tag{7b}$$

Here, we note that the constraint (7b) is the *partition matroid* constraint, which ensures that the optimal policy chooses only one member of $\mathcal{P}^i$ for each agent $i \in \mathcal{A}$ from the collective feasible set $\mathcal{P}$. The optimization problem is in the standard form of (3), which is known to be NP-hard [21]. The following result, whose proof is omitted due to space limitation, gives the cost of constructing the feasible set $\mathcal{P}$ and time complexity of solving optimization problem (7).

Lemma 3. (Time complexity of solving problem (7)). The cost of constructing the feasible set $\mathcal{P}$ for optimization problem (7) is of order $O(M\mathsf{D}^{\bar{N}_f})$, where $\mathsf{D} = \max(|\mathcal{N}_1|, \cdots, |\mathcal{N}_N|)$, $\bar{N}_f = \max\{|\mathsf{n}^i|\}_{\mathsf{n}^i \in \mathcal{P}}$ [1]. Furthermore, the time complexity of solving optimization problem (7) is $O(\prod_{i=1}^M \mathsf{D}^{\bar{N}_f^i})$ where $\bar{N}_f^i = \max\{|\mathsf{n}^i|\}_{\mathsf{n}^i \in \mathcal{P}^i}$. □

Our objective in this paper is to construct a sub-optimal policy to solve the persistent patrolling problem described above with a time complexity that is 'reasonable' (will be defined more preciously in the next section).

[1] With a slight abuse of notation, here we use $\mathsf{n}^i \in \mathcal{P}^i$ in place of $(\mathsf{n}^i, \mathbf{T}^i, i) \in \mathcal{P}^i$.

4. A SUB-MODULAR RECEDING HORIZON APPROACH TO PERSISTENT MONITORING

Lemma 3 indicates that the time complexity of finding an optimal patrolling policy increases exponentially by the horizon of the agents' policy N_f and also the number of agents M exploring the map. To reduce the computational cost, we can trade in optimality and divide the policy making horizon to multiple shorter horizon of length N_H and design an optimal policy for each horizon. In other words, in the policy making optimization problem (7), the search space $\mathcal{P}$ of the optimal policy is limited to sub-policies with the length of N_H. Even though this approach is able to cut down on the time complexity of optimization problem (7), the time complexity of finding an optimal policy still increases exponentially with the number of agents. We note here that in our problem of interest because of the changes that happen to various operational aspect of the problem, periodical solutions cannot be effective.

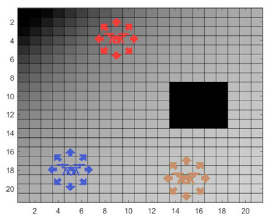
To provide a suboptimal policy making that is also robust to the online changes that can occur during the mission time, in what follows we propose a submodular receding horizon policy making algorithm for our persistent patrolling problem described in Section 3. To construct this algorithm, we first show that the reward function (6) is submodular over any given feasible policy set $\mathcal{P}$. The proof of the result below relies on several auxiliary results that are given in Appendix. The detailed proof of Theorem 4 is omitted for brevity and will appear elsewhere.

Theorem 4. (submodularity of reward function). Consider reward function defined in (4). Let $\mathcal{P}$ be a given set of policies of the form $\mathsf{p} = (\mathsf{n}_\mathsf{p}, \mathbf{T}_\mathsf{p}, i_\mathsf{p})$. Then, the reward function (6) is a monotone increasing and submadular set function over $\mathcal{P}$. □

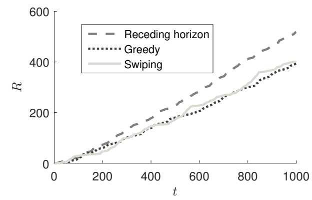
As we mentioned earlier, even though receding horizon approach is able to cut down on the time complexity of optimization problem (7), the time complexity of finding an optimal policy in each receding horizon N_H still increases exponentially with the number of agents (see Lemma 3). With the guarantee that Theorem 4 provides about the submodularity of the cost function of the optimization problem (7), we can now implement the submodular-based sequential greedy optimization Algorithm 1 to solve (7) in a polynomial time with the known lower bound of the sub-optimality given by Theorem 2. When implementing Algorithm 1, function g should be replaced by R. To implement the sequential greedy algorithm, we first calculate all the feasible policies $\mathcal{P}$ with length N_H for all the mobile agents $\mathcal{A}$. In each execution of the main loop of Algorithm 1, function $\mathsf{SGOpt}(\mathcal{P}, M)$ picks the sub-policy p that is most rewarding with consideration of all sub-policies chosen in previous execution of the loop (starting from $\bar{\mathcal{P}} = \emptyset$) and adds it to $\bar{\mathcal{P}}$. Then it eliminates all the sub-policies from the set $\mathcal{P}$ which are corresponding to the agent where p is originated from. This approach continues till there are no more sub-policies left in $\mathcal{P}$. After Algorithm 1 produces a solution, we implement the policy until a pre-specified event such as first agent arriving and completing the scan of its first designated geographical node, a mobile agent being added to the system, a mobile agent being removed from the system, a change in the Poisson process rate of a geographical node, or a change in topology including adding/deleting geographical nodes or edges between nodes happen.

5. DEMONSTRATIVE EXAMPLES

In this section, we use two numerical examples to demonstrate some of the properties of our algorithms. In these



(a) λ^i map where the darker area shows higher probability of detecting an event.



(b) The utility of Receding horizon algorithm compared to Greedy and Swiping algorithm

Fig. 2. The simulation scenario and results of example 1: three agents are monitoring a 21×21 grid with non-uniform Poisson rate.

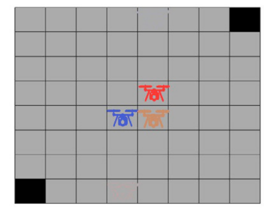
examples, we assume that the geographical area of interest is partitioned into uniform cells in which the center of the cells are the geographical nodes constituting $\mathcal{V}$. Each node is connected to the neighboring nodes via vertical, horizontal and diagonal edges. For simplicity, we assume that the travel time and the scan time of all the mobile agents are the same. We also assume that the travel time along all the edges is the same.

In the first example, we consider a 21×21 grid shown in Fig. 2(a). The heat map on the gird demonstrates the event rate λ_i of each cell which ranges over $[0.0025, 0.01]$ (the higher intensity of the cells indicates higher value of λ_i). We assume that there is a 5×5 mesh with a high rate of 0.03 (shown by black region on the grid). We change the location of this important region 10 times, randomly over the full horizon of the simulation. This scenario is designed to consider a case that the event rate in the map changes with time. In this simulation, we use 3 agents to patrol over the area with the objective of scoring maximum reward (6) for the team. Since the optimal solution is NP-hard, we consider three alternative methods of patrolling: a submadular receding horizon approach of Section 4 with $N_H = 4$, the swiping method in which the area is partitioned equally between the agents to sweep, and finally the greedy method in which each agent chooses the most rewarding neighboring cell and moves there. Figure 2(b) shows the total reward scored by the agents using these three different methods. As we can see, the submadular receding horizon approach results in a better policy that gathers more reward, i.e., detects more events.

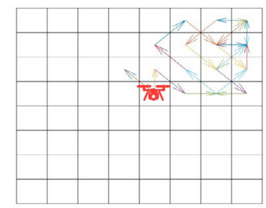
In the second example, we consider an 8×8 grid shown in Fig. 3(a). As shown on the figure, we assume that the Poisson rate of event detection in all the cells except the two in the up-right and bottom-left corners have uniform value 0.01 and the corners have values of 4 and 2 respectively. The initial location of the three mobile agents used in this simulation is shown in Fig. 3(a). Figures 3(b)-(d) show the trajectories of the agents as they follow the dispatch commands of the submodular receding horizon approach of Section 4 with $N_H = 5$ over 50 planning events. As these figures show, our proposed approach is spreading the agents in an efficient manner of two of the agents each patrolling the areas near the cells with high rate of event occurring and the third agent exploring over the remaining area. This behavior is aligned with intuitive expectation from an optimal patrolling policy.

6. CONCLUSION AND FUTURE WORK

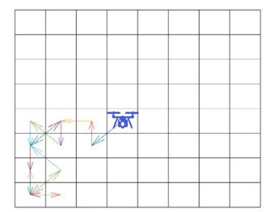
We proposed a solution for a persistent monitoring of a finite number of inter-connected geographical nodes in an urban environment with the purpose of maximizing the expected value of event detection. We modeled the probability of discovering at least one event in each geographical node as a Poisson distribution and tied this with trajectory scheduling of the agents via a utility function. We showed that maximizing the utility function is NP-hard. We also argued that for our problem of interest, a one-shot optimization is not robust to online changes of the system. Hence, we proposed a submodular receding horizon scheme to induce robustness and also decrease the computational cost of dispatch policy design. Our proposed method is a suboptimal solution with a known lower bound of optimality gap. Our future work is to consider decentralized implementation of our proposed algorithm and also to address the shortsightedness of the receding horizon approach.



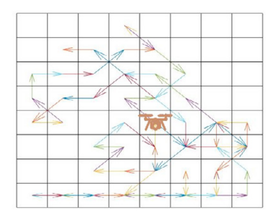
(a) Grid map of example 2



(b) The trajectory of the red agent



(c) The trajectory of the blue agent



(d) The trajectory of the brown agent

Fig. 3. Grid map of example 2 and the trajectories of the three mobile agents: the submodular receding horizon approach of Section 4 is used to dispatch the agents over 50 planning steps. The arrows show the directions that the agents traveled.

REFERENCES

[1] S. Karlin, *A first course in stochastic processes*. Academic Press, 2014.

[2] C. Yuan, Y. Zhang, and Z. Liu, “A survey on technologies for automatic forest fire monitoring, detection, and fighting using unmanned aerial vehicles and remote sensing techniques,” *Canadian Journal of Forest Research*, vol. 45, no. 7, pp. 783–792, 2015.

[3] N. Henry and O. Henry, “Wireless sensor networks based pipeline vandalisation and oil spillage monitoring and detection: main benefits for nigeria oil and gas sectors,” *The SIJ Transactions on Computer Science Engineering & its Applications (CSEA)*, vol. 3, no. 1, pp. 1–6, 2015.

[4] R. Engler, A. Guisan, and L. Rechsteiner, “An improved approach for predicting the distribution of rare and endangered species from occurrence and pseudo-absence data,” *Journal of Applied Ecology*, vol. 41, no. 2, pp. 263–274, 2004.

[5] A. Machado, G. Ramalho, J. Zucker, and A. Drogoul, “Multi-agent patrolling: An empirical analysis of alternative architectures,” in *International Workshop on Multi-Agent Systems and Agent-Based Simulation*, pp. 155–170, Springer, 2002.

[6] A. Almeida, G. Ramalho, H. Santana, P. Tedesco, T. Menezes, V. Corruble, and Y. Chevaleyre, “Re-

cent advances on multi-agent patrolling," in *Brazilian Symposium on Artificial Intelligence*, pp. 474–483, Springer, 2004.
[7] Y. Chevaleyre, "Theoretical analysis of the multi-agent patrolling problem," in *Intelligent Agent Technology*, pp. 302–308, IEEE, 2004.
[8] F. Pasqualetti and A. F. F. Bullo, "On cooperative patrolling: Optimal trajectories, complexity analysis, and approximation algorithms," *IEEE Transactions on Robotics*, vol. 28, no. 3, pp. 592–606, 2012.
[9] K. Yu, S. Karaman, and D. Rus, "Persistent monitoring of events with stochastic arrivals at multiple stations," *IEEE Transactions on Robotics*, vol. 31, no. 3, pp. 521–535, 2015.
[10] M. Donahue, G. Rosman, K. Kotowick, D. Rus, and C. Baykal, "Persistent surveillance of transient events with unknown statistics," tech. rep., MIT Lincoln Laboratory Lexington United States, 2016.
[11] Y. Khazaeni and C. Cassandras, "Event-driven trajectory optimization for data harvesting in multi-agent systems," *IEEE Transactions on Control of Network Systems*, 2017.
[12] N. Zhou, X. Yu, S. Andersson, and C. Cassandras, "Optimal event-driven multi-agent persistent monitoring of a finite set of data sources," *IEEE Transactions on Automatic Control*, 2018.
[13] S. Chen, F. Wu, L. Shen, J. Chen, and S. Ramchurn, "Multi-agent patrolling under uncertainty and threats," *PloS One*, vol. 10, no. 6, p. e0130154, 2015.
[14] A. Farinelli, L. Iocchi, and D. Nardi, "Distributed on-line dynamic task assignment for multi-robot patrolling," *Autonomous Robots*, vol. 41, no. 6, pp. 1321–1345, 2017.
[15] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies," *Journal of Machine Learning Research*, vol. 9, no. Feb, pp. 235–284, 2008.
[16] T. Summers, F. Cortesi, and J. Lygeros, "On submodularity and controllability in complex dynamical networks.," *IEEE Transactions Control of Network Systems*, vol. 3, no. 1, pp. 91–101, 2016.
[17] Z. Liu, A. Clark, P. Lee, L. Bushnell, D. Kirschen, and R. Poovendran, "Submodular optimization for voltage control," *IEEE Transactions on Power Systems*, vol. 33, no. 1, pp. 502–513, 2018.
[18] A. Clark, P. Lee, B. Alomair, L. Bushnell, and R. Poovendran, "Combinatorial algorithms for control of biological regulatory networks," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 2, pp. 748–759, 2018.
[19] A. Krause and C. Guestrin, "Near-optimal observation selection using submodular functions," in *AAAI*, vol. 7, pp. 1650–1654, 2007.
[20] A. Clark, L. Bushnell, and R. Poovendran, "A supermodular optimization framework for leader selection under link noise in linear multi-agent systems," *IEEE Transactions on Automatic Control*, vol. 59, no. 2, pp. 283–296, 2014.
[21] B. Gharesifard and S. Smith, "Distributed submodular maximization with limited information," *to appear in IEEE Transactions on Control of Network Systems*, 2018.
[22] G. Nemhauser, L. Wolsey, and M. Fisher, "An analysis of approximations for maximizing submodular set functionsi," *Mathematical programming*, vol. 14, no. 1, pp. 265–294, 1978.
[23] L. Lovász, "Submodular functions and convexity," in *Mathematical Programming The State of the Art*, pp. 235–257, Springer, 1983.
[24] Z. Kadelburg, D. Dukic, M. Lukic, and I. Matic, "Inequalities of karamata, schur and muirhead, and some applications," *The Teaching of Mathematics*, vol. 8, no. 1, pp. 31–45, 2005.
[25] R. Karp, "Reducibility among combinatorial problems," in *Complexity of computer computations*, pp. 85–103, Springer, 1972.

Appendix A.

Definition A.5. The non increasing sequence $(\delta t)_1^n$ majorizes the non increasing sequence $(\delta v)_1^n$, if

- $\delta t_1 \geq \delta t_2 \geq ... \geq \delta t_n$ and $\delta v_1 \geq \delta v_2 \geq ... \geq \delta v_n$,
- $\delta t_1 + ... + \delta t_i \geq \delta v_1 + ... + \delta v_i$ for $\forall i \in \{1, ..., n-1\}$,
- $\delta t_1 + ... + \delta t_n = \delta v_1 + ... + \delta v_n$.

Theorem A.6. (Karamata's inequality [24]). Given a real function $f : \mathbb{R} \to \mathbb{R}$, and sequences $(\delta t)_1^n$ and $(\delta v)_1^n$ where $(\delta t)_1^n$ majorizes $(\delta v)_1^n$, we have

$$f(\delta t_1) + ... + f(\delta t_n) \leq f(\delta v_1) + ... + f(\delta v_n).$$

□

We develop the following results to use in the proof Theorem 4. Due to space limitation, the proofs these results are not include here.

Corollary A.7. Given $f : \mathbb{R} \to \mathbb{R}$ with $f(0) = 0$, and sequences $(\delta t)_1^n$ and $(\delta v)_1^m$ with $n \leq m$, where for $(\delta t)_1^n$ and $(\delta v)_1^m$ we have

$$\delta t_1 + ... + \delta t_i \geq \delta v_1 + ... + \delta v_i, \quad \forall i \in \{1, ..., n-1\}$$
$$\delta t_1 + ... + \delta t_n = \delta v_1 + ... + \delta v_m.$$

Then

$$f(\delta t_1) + ... + f(\delta t_n) \leq f(\delta v_1) + ... + f(\delta v_m). \quad (A.1)$$

□

Corollary A.8. Given a monotone increasing and concave function $f : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, and $0 \leq a \leq c$ and $0 \leq b \leq d$ then

$$f(c) + f(d) - f(c + d) \geq f(a) + f(b) - f(a + b). \quad (A.2)$$

□

Lemma A.9. For any $(q)_1^l$, let

$$g((q)_1^l) = \sum_{i=1}^{l-1} f(\Delta q_i), \quad (A.3)$$

where $\Delta q_i = q_{i+1} - q_i$ and $f \in \mathcal{K}$ is a concave and semi positive and increasing function with $f(0) = 0$. Now, consider two increasing sequences $(t)_1^n$ and $(u)_1^l$ and concatenation $(a)_1^{n+l} = (t)_1^n \oplus (u)_1^l$, then

$$g((a)_1^{n+l}) - g((t)_1^n) \geq 0.$$

□

Lemma A.10. For any $(q)_1^l$, let

$$g((q)_1^l) = \sum_{i=1}^{l} f(\Delta q_i), \quad (A.4)$$

where $\Delta q_i = q_{i+1} - q_i$ and $f \in \mathcal{K}$ is a concave and positive and increasing function with $f(0) = 0$. Now, consider three increasing sequences $(t)_1^n$ and $(v)_1^m$ and $(u)_1^l$ and concatenations $(a)_1^{n+l} = (t)_1^n \oplus (u)_1^l$ and $(b)_1^{m+l} = (v)_1^m \oplus (u)_1^l$ where $(v)_1^m$ is a sub-sequence of $(t)_1^n$, then

$$\left(g((b)_1^{m+l}) - g((v)_1^m)\right) - \left(g((a)_1^{n+l}) - g((t)_1^n)\right) \geq 0.$$