Cloud-Supported Coverage Control for Persistent Surveillance Missions *

Jeffrey R. Peters[†]

Department of Mechanical Engineering Department of Mechanical Engineering University of California Santa Barbara, CA 93106 Email: jrpeters@engr.ucsb.edu

Sean Wang

University of California Santa Barbara, CA 93106 Email: seanwang@umail.ucsb.edu

Amit Surana Systems Department United Technologies Research Center East Hartford, CT 06118 Email: suranaa@utrc.utc.com

Francesco Bullo Department of Mechanical Engineering University of California Santa Barbara, CA 93106 Email: bullo@engr.ucsb.edu

A cloud-supported coverage control scheme is proposed for multi-agent, persistent surveillance missions. This approach decouples assignment from motion planning operations in a modular framework. Coverage assignments and surveillance parameters are managed on the *cloud*, and transmitted to mobile agents via unplanned and asynchronous exchanges. These updates promote load-balancing, while also allowing effective pairing with typical path planners. Namely, when paired with a planner satisfying mild assumptions, the scheme ensures that (i) coverage regions remain connected and collectively cover the environment, (ii) regions may go uncovered only over bounded intervals, (iii) collisions (sensing overlaps) are avoided, and (iv) for time-invariant event likelihoods, a Pareto optimal configuration is produced in finite time. The scheme is illustrated in simulated missions.

1 Introduction

1.1 **Cloud-Supported Multi-Agent Surveillance**

Autonomous sensors are used in many modern surveillance missions, including search and rescue [1], environmental monitoring [2], and military reconnaissance [3]. Such missions often require agents to periodically exchange data with a central *cloud* (repository) and, when operating in nonideal environments or under hardware limitations, these po-

[†]Corresponding author

tentially sporadic exchanges may be the only means of sharing real-time information across agents. For example, autonomous underwater vehicles are virtually isolated due to difficulties in underwater data transfer, and rely on periodic surfacing to communicate with a tower [4]. Other applications that may have this constraint include data mules that periodically visit ground robots [5], and supervisory missions that require unmanned vehicles to send data to a remotely located operator [6]. Such scenarios require robust and flexible frameworks for real-time autonomous coordination.

Single-agent surveillance strategies range from simple a priori tour construction [7] to more complex methods involving Markov chains [8], optimization [9], or Fourier analysis [10]. However, it is not straightforward to generalize single-agent approaches for multi-agent missions: Naive approaches where each agent follows an independent policy often result in poor performance and introduce collision risks, while other generalizations may require joint optimizations that are intractable for even modestly sized problems [11]. Distributed control can sometimes alleviate scaling issues; however, such setups typically rely on ideal peer-to-peer data transfer making them ill-posed in many cloud-based setups.

In contrast, decomposition-based approaches to multiagent surveillance, which decouple the assignment and routing problem by dividing the agent workspace, are popular in practice as they offer a straightforward, modular framework to reasonably accomplish surveillance goals, despite sacrificing optimality. However, in cloud-based architectures that rely solely on agent-cloud exchanges for real-time data transfer, implementation of such an approach is not straightforward using existing approaches to dynamic workspace de-

^{*}This work has been sponsored by the U.S. Army Research Office and the Regents of the University of California, through Contract Number W911NF-09-D-0001 for the Institute for Collaborative Biotechnologies, and that the content of the information does not necessarily reflect the position or the policy of the Government or the Regents of the University of California, and no official endorsement should be inferred.



Fig. 1. When complete or pairwise coverage updates are impossible, two updates are required to move from the left-most to the right-most configuration. The red region is updated first, introducing a collision (redundant sensing) risk.

composition. Indeed, in this case, updated mission information is only relayed to one agent at a time, rendering traditional partitioning schemes, which rely on complete or pairwise coverage updates, impossible, and existing strategies that utilize single-agent updates, e.g., [12], may introduce undesirable configurations or collision-risks (Fig. 1).

This work extends existing literature by introducing a cloud-supported, decomposition-based framework for multiagent persistent surveillance that promotes effective coverage without introducing collision (redundant sensing) risks and without requiring ideal or pre-planned data exchanges. As such, the proposed framework allows agents to effectively respond to changes in the environment (spatial event likelihood) without having to be collected and re-deployed due to the inability to transmit data to all agents simultaneously. That is, when mission goals (captured by the event likelihood) change after agents are already in the field, the cloud incrementally relays the updated information through sporadic exchanges with individual agents, driving them to a new coverage configuration without introducing undesirable intermediate states. This framework also allows the cloud to act as a data fusion center if necessary, dynamically combining local sensory information collected by individual agents.

1.2 Related Literature

Multi-agent coverage control problems have generated extensive research. Typical strategies involve optimization [13], auctions [14], meta-heuristics [15], potential fields [16], Markov decision processes [17], among others [18]. Of particular relevance is multi-agent persistent surveillance (monitoring), in which a sensor team is tasked with continual surveillance of some region, requiring subregions to be visited multiple (or infinitely many) times to minimize a cost, e.g., the time between visits or the likelihood of detecting events [11]. Persistent surveillance is a generalization of *patrolling*, where agents follow closed tours to protect or supervise an environment. Most current solutions to patrolling problems utilize operations research, non-learning multi-agent systems, and multi-agent learning [19]; however, formulations are often one-dimensional and solutions usually reduce to "back and forth" motions that do not readily extend to general scenarios, e.g., [20].

The framework herein employs a type of workspace decomposition in constructing solutions to the multi-agent problem. In the context of general robotic applications, the term *workspace decomposition* refers to any number of strategies that are employed in order to represent a complex workspace or solution space in a simpler, often lowerdimensional form. For robotic motion planning involving obstacles, this often involves the representation of the free configuration space in a graphical or modular form that captures essential connectivity properties, e.g., typical roadmap and cellular decomposition planning methods take this approach [21, 22]. In multi-agent applications, workspace decomposition methods can also be used to reduce the complexity of a problem through the assignment of sub-tasks to individual agents, i.e., for task assignment [23]. This is the approach taken herein to reduce the multi-agent problem into a set of single-agent problems. Since the assignment problem is often difficult to solve (it has been shown to be NP-hard in some domains [23]), multi-agent planning solutions based on this type of assignment usually lead to sub-optimal solutions. Despite this drawback, assignmentbased approaches remain popular in practice due to their simplicity and scalability [11]. Outside of workspace decomposition, other strategies for reducing multi-agent planning problems into a set of single-agent problems include naive approaches, where each agent independently determines its own actions by locally solving a complete problem over the full workspace, and, provided sufficiently reliable sensing and data-transfer capabilities, distributed approaches, in which agents each solve a sub-problem based on information shared over a communication network [24].

For planar persistent surveillance, decomposition-based approaches typically consist of two primary components: partitioning and single-agent routing. The most common approaches to optimal partitioning in convex environments are based on Voronoi partitions [25], and effective schemes exist for constructing centroidal Voronoi, equitable, or other types of optimal partitions under communication, sensing, and workload constraints [26-28]. Non-convex workspaces are typically addressed by representing the environment as a graph, on which a number of graph partitioning schemes can be used [29]. In robotics, discrete partitioning is often considered under communication constraints, e.g., pairwise gossip [30] or asynchronous one-to-base station communication [12]. Our proposed scheme most closely mirrors [12], in which agents communicate sporadically with a base station; however, our approach employs additional logic to ensure the resultant coverage regions retain properties that are consistent with effective, decomposition-based surveillance.

Single-agent path planners for persistent surveillance typically operate on graphs [31, 32], and classical problems, e.g., the Traveling Salesperson Problem [33], often play a key role. Stochasticity can be introduced using tools such as Markov chains [8]. Schemes for non-discrete spaces (open subsets of Euclidean space), are less common. Here, strategies include *a priori* construction of motion routines [34], adaptation of static coverage strategies [35], the use of random fields [36], and spectral decomposition [10]. The modular framework herein incorporates any single-agent planner satisfying mild assumptions (see Section 5).

Remarkably few papers explicitly consider the implications of combining dynamic partitioning with continuous routing for multi-agent persistent surveillance. Existing research is mostly preliminary, considering ideal data transfer and simplistic methods. The authors of [37] employ a sweeping algorithm for partitioning and guide vehicle motion via lawn-mower patterns, while [38] uses rectangular partitions and a reactive routing policy which, in ideal cases, reduces to spiral search patterns. The authors of [39] use slightly more sophisticated partitioning in tandem with lawn-mower trajectories. In [40], partitions are based on the statistical expectation of target presence, but ideal communication is assumed. Others, e.g., [41], employ decomposition-based structures, but focus on task-assignment without detailed treatment of the combined assignment/routing protocol.

Our work uses a cloud-supported computational framework. Cloud-based robotic infrastructures (cloud robotics) have generated growing research interest, as they can provide many benefits to complex systems, such as the storage and analysis of "big data," the availability of parallel grid computing, the potential for collective learning, and the utilization of human computation [42]. In multi-agent systems, cloud-supported schemes have been used for tasks such as collective optimization [43], rendezvous [44], and coordinated task-assignment [12]. Our use of the cloud-based architecture is primarily motivated by supervisory systems involving unmanned vehicles. Here, this architecture arises naturally, since mobile agents are required to transmit sensor data to a remotely located human operator for analysis (thus requiring a central repository), and harsh operational environments often prohibit reliance on peer-to-peer communication. However, the proposed framework is suitable for use in any similar setup, and can also run in parallel with other cloud-supported operations (e,g, data analysis, learning, etc.) within complex missions.

1.3 Contributions

This work develops a cloud-supported, decompositionbased, multi-agent coverage control framework for persistent surveillance, which relies only on sporadic, unscheduled exchanges between agents and a central cloud for data transfer. In particular, we develop a sophisticated partitioning and coordination scheme that can be effectively paired with singleagent trajectory planners. This naturally leads to the complete, modular framework in which high-level coverage is coordinated on the cloud and agent trajectories are generated independently via on-board planners. We encompass realistic constraints including restrictive communication, dynamic environments, and non-parametric event likelihoods.

Specifically, our dynamic partitioning scheme only requires agents to sporadically upload and download data from the cloud. The cloud runs updates to govern region assignments, while also manipulating high-level surveillance parameters. We prove that this update structure has many desirable properties: coverage regions collectively form a connected *m*-covering and evolve at a time-scale that allows for appropriate agent reaction, no subregion remains uncovered indefinitely, local likelihood functions have disjoint support, among others. For certain cases, we show that the set of coverage regions and associated generators converges to a Pareto optimal pair in finite time. We show that the combination of our partitioning scheme with a generic trajectory planner ensures collision (sensing overlap) avoidance, provided the planner obeys natural restrictions. Finally, we illustrate our framework through numerical examples.

Our partitioning scheme is primarily motivated by [12]; however, the algorithms herein are explicitly designed to operate within a multi-agent surveillance framework and introduce additional logic parameters to evoke a set of desirable geometric and temporal properties. Our proposed scheme has the following advantages: First, our framework considers a modified cost function that uses subgraph distances to maintain connectivity of intermediate coverage regions, ensuring that agents can visit their entire assigned region without entering another agent's territory. Second, timing parameters are manipulated to provide inherent collision (redundant sensing) avoidance when the scheme is paired with low-level motion planners (Fig. 2). Third, our algorithms explicitly manipulate local likelihood functions maintained by the agents to guarantee that each has support within an allowable region, promoting seamless and modular pairing with any trajectory planner that uses the support of the event likelihood to govern agent routes, e.g., [10]. The framework has these features while simultaneously maintaining similar convergence properties as the algorithms in [12].

For clarity and readability in what follows, we postpone all theorem proofs until the Appendix.

2 Mission Overview and Solution Approach

A team of m mobile agents¹, each equipped with an onboard sensor, is tasked with persistently monitoring a nontrivial, planar region of interest. The primary goal of the mission is to collect sensor data about some dynamic event or characteristic, e.g., an intruder. Collected data is periodically uploaded to the cloud. Agents must move within the environment to obtain complete sensory information. Ideally, agent motion should be coordinated so that:

- 1. the sensing workload is balanced across agents,
- 2. no subregion goes unobserved indefinitely,
- 3. agents never collide (have sensor overlap), and
- 4. the search is biased toward regions of greater interest.

To achieve these goals, we propose a decomposition-based approach in which each agent's motion is restricted to lie within a dynamically assigned *coverage region*. The partitioning component (operating on the cloud), defines these coverage regions and provides high-level restrictions on agent motion through the manipulation of surveillance parameters, while the trajectory planning component (operating on-board each agent) incrementally constructs agent motion paths. We assume only asynchronous, cloud-agent data transfer, i.e., agents sporadically exchange data with the cloud, where inter-exchange times are not specified *a priori*, but are subject to an upper bound.

Broadly, our framework operates as follows (Fig. 2). Initial coverage variables are communicated to the agents prior

¹Each agent is uniquely paired with a coverage region, so the quantity m represents both the number of agents and the number of regions.



Fig. 2. Illustration of the proposed strategy. The partitioning component (executed on the cloud) manages coverage regions and introduces logic to prevent collisions, while the trajectory planning component (executed onboard each agent) governs agent motion.

to deployment, i.e., relevant initial information is known to each agent at the mission onset. Once in the field, agents communicate sporadically with the cloud. During each agent-cloud exchange, the cloud calculates a new coverage region solely for the communicating agent, along with a set of timing and surveillance parameters that serve to govern the agent's high-level motion behavior, and transmits the update. The update algorithm also alters relevant global variables maintained on the cloud. Once the update completes, the data-link is terminated and the agent follows the trajectory found via its onboard planner. Notice that this structure is a type of *event-triggered* control, since high-level updates only occur in the event of an agent-cloud exchange.

3 Problem Setup

The cloud, as well as each agent, has its own local processor. "Global" information is stored on the cloud, while each agent only stores information pertinent to itself.

Convention 1. Subscripts $i, j, or \ell$ denote an entity or set element relevant to agent $i, j, or \ell$, resp. The superscript 'A' denotes an entity that is stored by the agent's local processor.

A storage summary is shown in Table 1. We expand on these, and define other relevant mathematical constructs here.

3.1 Agent Dynamics

Each agent (sensor) *i* is a point mass that moves with maximum possible speed $s_i > 0$. Define $s := \{s_i\}_{i=1}^m$. Notice that this setup allows heterogeneity with respect to speed, i.e., agents can travel at different speeds.

3.2 Communication Protocol

Each agent periodically exchanges data with the cloud. Assume the following:

- each agent can identify itself to the cloud, and transmit/receive data,
- 2. there is a lower bound $\underline{\Delta} > 0$ on the time between any two successive exchanges involving the cloud, and
- 3. there is an upper bound $\Delta > 0$ on the time between any *single agent's* successive exchanges with the cloud.

Assume that agent-cloud exchanges occur instantaneously, and notice that condition 2 implies no two exchanges (involving any agents) occur simultaneously². Since computation time is typically small in comparison to inter-exchange times and exchanges occur sporadically, these assumptions are without significant loss of generality.

3.3 Environment

Consider a bounded surveillance environment as a finite grid of disjoint, non-empty, simply-connected subregions. We represent the grid as a weighted graph G(Q) := (Q, E), where Q is the set of vertices (each representative of a unique grid element), and E is the edge set comprised of undirected, weighted edges $\{k_1, k_2\}$ spanning vertices representing *adjacent*³ grid elements. Let the weight associated to $\{k_1, k_2\}$ be some finite upper bound on the minimum travel distance between any point in the grid element associated k_1 to any point in the grid element associated to k_2 along a path that does not leave the union of the two elements. Locations of environmental obstacles and prohibited areas are known and are not included in the graphical representation G(Q).

Consider $Q' \subseteq Q$. A vertex $k_1 \in Q$ is *adjacent* to Q'if $k_1 \notin Q'$ and there exists $\{k_1, k_2\} \in E$ with $k_2 \in Q'$. Define G(Q') := (Q', E') as the subgraph of G(Q) induced by the vertex set Q'. A *path* on G(Q') between $k_1, k_n \in Q'$ is a sequence $(k_1, k_2, ..., k_n)$, where $k_1, k_2, ..., k_n \in Q'$ and $\{k_r, k_{r+1}\} \in E'$ for $r \in \{1, ..., n-1\}$. We say Q' is *connected* if a path exists in G(Q') between any $k_1, k_2 \in Q'$. Let $d_{Q'} :$ $Q' \times Q' \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ be the standard distance on G(Q'), i.e., the length of a shortest weighted path in G(Q') (if none exists, $d_{Q'}$ takes value ∞). Notice that $d_Q(k_1, k_2) \leq d_{Q'}(k_1, k_2)$ for any $k_1, k_2 \in Q'$. With slight abuse of notation, we also let $d_{Q'}$ denote the map $d_{Q'} : Q' \times 2^{Q'} \to \mathbb{R}_{\geq 0} \cup \{\infty\}$, where $d_{Q'}(k, Q'')$ is the length of a shortest weighted path in G(Q')between k and any vertex in Q''.

3.4 Coverage Regions

An *m*-covering of Q is a family $P := \{P_i \subseteq Q\}_{i=1}^m$ satisfying (i) $\bigcup_{i=1}^m P_i = Q$, and (ii) $P_i \neq \emptyset$ for all *i*. Define $\operatorname{Cov}_m(Q)$ as the set of all possible *m*-coverings of Q. An *m*-partition of Q is an *m*-covering that also satisfies (iii) $P_i \cap P_j = \emptyset$ for all $i \neq j$. An *m*-covering or *m*-partition P is connected if each P_i is connected. In what follows, the cloud maintains an *m*-covering P of Q, and surveillance responsibilities are assigned by pairing each agent *i* with $P_i \in P$ (called agent *i*'s coverage region). Each agent maintains a copy P_i^A of P_i . The cloud also stores a set $c := \{c_i \in Q\}_{i=1}^m$ (c_i is the generator of P_i), and each agent *i* maintains a copy c_i^A of c_i .

3.5 Identifiers, Timers, and Auxiliary Variables

The proposed algorithms introduce logic and timing variables to ensure an effective overall framework. To each $k \in Q$, assign an *identifier* $ID_k \in \{1, ..., m\}$. Define $I\mathcal{D} := \{ID_k\}_{k \in Q}$, and let $P^{I\mathcal{D}} := \{P_i^{I\mathcal{D}}\}_{i=1}^m$, where $P_i^{I\mathcal{D}} := \{k \in Q \mid k \in Q \mid k$

²Mathematically, the bound $\underline{\Delta}$ also prevents *zeno behavior*.

³Travel between the elements without leaving their union is possible.

ID_k = i}. Notice $P^{I\mathcal{D}}$ is an *m*-partition of *Q*. For each agent *i*, define a timer T_i having dynamics $\dot{T}_i = -1$ if $T_i \neq 0$, and $\dot{T}_i = 0$ otherwise. Define $T := \{T_i\}_{i=1}^m$. Each agent *i* maintains a local timing variable τ_i^A . Even though τ_i^A plays a similar role to T_i , note that τ_i^A is constant unless explicitly updated, while T_i has autonomous dynamics. Next, the cloud maintains a set $\omega := \{\omega_i\}_{i=1}^m$, where ω_i is the time of agent *i*'s most recent exchange with the cloud. Each agent maintains a copy ω_i^A of ω_i . Finally, each agent stores a subset $P_i^{A,pd} \subseteq P_i^A$ which collects vertices that have recently been added to P_i^A .

3.6 Likelihood Functions

The likelihood of a relevant event occurring within any subset of the surveillance region is maintained on the cloud in the form of a time-varying probability mass function⁴ Φ : $Q \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{>0}$. For simplicity, assume that, at any *t*, the instantaneous support, supp $(\Phi(\cdot, t))$, equals *Q*.

Define each agent's *local likelihood* $\Phi_i^A : Q \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ as the function that, loosely, represents the agent's local belief regarding events. Specifically, define

$$\Phi_i^A(k,t) = \begin{cases} \text{if } k \in P_i^A \text{ and} \\ \Phi(k,t), & \left(t - \omega_i^A \ge \tau_i^A \text{ or } k \notin P_i^{A,\text{pd}}\right), \\ 0, & \text{otherwise.} \end{cases}$$
(1)

The conditions defining Φ_i^A are understood as follows: at some time *t*, an element $k \in Q$ only belongs to $\operatorname{supp}(\Phi_i^A(\cdot,t))$ if (i) $k \in P_i^A$, and (ii) sufficient time has passed since *k* was first added to P_i^A , as determined by the parameters τ_i^A , ω_i^A , and $P_i^{A, \text{pd}}$. In general, each Φ_i^A will be different⁵ from Φ .

Remark 1 (Global Data). If global knowledge of Φ is not available instantaneously to agent i, Φ_i^A can alternatively be defined by replacing $\Phi(k,t)$ in (1) by $\Phi(k,\omega_i^A)$. All subsequent theorems hold under this alternative definition.

Remark 2 (Data Storage). The cost of storing a graph as an adjacency list is O(|Q| + |E|). The generator set c, each element of P, and the identifier set $I\mathcal{D}$ are stored as integral vectors. The timer set T and the set ω are are stored as real vectors, while Φ is stored as a time-varying real vector. Thus, the cost of storage on the cloud is O(m|Q| + |E|). Similarly, each agent's local storage cost is O(|Q| + |E|).

4 Dynamic Coverage Update Scheme

Adopt following convention for the remaining analysis.

Convention 2. Suppose that:

- 1. $\min \emptyset := \max \emptyset := 0$, and
- 2. given a specific time instant, superscripts '-' or '+' refer to a value before and after the instant in question, resp.

	Stored on the Cloud	
Variable	Description	
G(Q)	Graphical representation of the environment	
Р	<i>m</i> -covering of Q ($P \in Cov_m(Q)$)	
С	Set of generators $(c \in Q^m)$	
$I\mathcal{D}$	Set of identifiers $\left(I\mathcal{D} \in \{1,\ldots,m\}^{ \mathcal{Q} }\right)$	
Т	Set of Timers $(T \in \mathbb{R}^m_{\geq 0})$	
ω	Set of most recent exchange times $(\omega \in \mathbb{R}^m_{\geq 0})$	
Φ	Global likelihood $(\Phi: Q imes \mathbb{R}_{\geq 0} o \mathbb{R}_{> 0})$	
Stored by Agent <i>i</i>		
Variable	Description	
G(Q)	Graphical representation of the environment	
P_i^A	Coverage region $(P_i^A \subset Q)$	
c_i^A	Coverage region generator $\left(c_i^A \in Q\right)$	
$P_i^{A,\mathrm{pd}}$	Set of "recently added" vertices $\left(P_i^{A, \text{pd}} \subseteq P_i^A\right)$	
$ au^A_i$	Local timing parameter $(\tau_i^A \in \mathbb{R})$	
ω_i^A	Most recent exchange time $(\omega_i^A \in \mathbb{R}_{\geq 0})$	
Φ^A_i	Local likelihood $\left(\Phi^A_i: Q imes \mathbb{R}_{\geq 0} o \mathbb{R}_{\geq 0} ight)$	

Table 1. Storage Summary

4.1 Additive Set

We start with a definition.

Definition 1 (Additive Set). Given $k \in P_i^{I\mathcal{D}}$, the additive set $P_i^{\text{add}}(k) \subseteq Q$ is the largest connected subset satisfying:

1. $P_i^{I\mathcal{D}} \subseteq P_i^{add}(k)$, and 2. for any $h \in P_i^{add}(k) \cap P_j$, where $j \neq i$: (a) $T_j = 0$, and

(b)
$$\frac{1}{s_i} d_{P_i^{\text{add}}(k)}(h,k) < \frac{1}{s_j} d_{P_j}(h,c_j).$$

The following characterizes well-posedness of Definition 1.

Proposition 1 (Well-Posedness). If $P_i^{I\mathcal{D}}$ is connected and disjoint from $\bigcup_{j \neq i} P_j$, then $P_i^{add}(k)$ exists and is unique for any $k \in P_i^{I\mathcal{D}}$.

Proof. With the specified conditions, $P_i^{I\mathcal{D}}$ is connected and satisfies 1 and 2 in Definition 1; $P_i^{\text{add}}(k)$ is the unique, maximally connected superset of $P_i^{I\mathcal{D}}$ satisfying 1 and 2.

Under the conditions of Proposition 1, if $h \in P_i^{\text{add}}(k)$, then $\max\{T_j \mid j \neq i, h \in P_j\} = 0$ and there is a path from *k* to *h* in $G(P_i^{\text{add}}(k))$ that is shorter than the optimal path spanning c_j and *h* within $G(P_i)$, for any $j \neq i$ with $h \in P_j$.

 $^{{}^{4}\}Phi(\cdot,t)$ is a probability mass function for any *t*.

 $^{{}^{5}\}Phi_{i}^{A}$ is not normalized and thus may not be a time-varying probability mass function in a strict sense.

4.2 Cloud-Supported Coverage Update

Define a cost function $\mathcal{H}: Q^m \times \operatorname{Cov}_m(Q) \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ by

$$\mathcal{H}(c,P,t) = \sum_{k \in Q} \min_{i} \left\{ \frac{1}{s_i} d_{P_i}(k,c_i) \, | \, k \in P_i \right\} \Phi(k,t)$$

If (i) each agent is solely responsible for events within its coverage region, and (ii) events occur proportionally to Φ , then $\mathcal{H}(c, P, t)$ is understood as the expected time required for an agent to reach a randomly occurring event from its region generator at time t; related functions are studied in [12,26,30]. Algorithm 1 defines the operations performed on the cloud when agent *i* makes contact at time t_0 . Here,

Algorithm 1: Cloud-Supported Coverage Update **Data**: $t_0, P, c, \Phi, \omega, \overline{\Delta}, \Delta_{\rm H}, T, I\mathcal{D}, s$ **Result**: $P, c, P_i^A, c_i^A, P_i^{A, \rm pd}, T, \tau_i^A, \Phi_i^A, \omega, \omega_i^A, I\mathcal{D}$ begin % Initialize, remove regions others have claimed Initialize $P^* = P^{\text{test}} = P$, $c^* = c^{\text{test}} = c$ 1 Set $P_i^* = P_i^{\text{test}} = P_i^{I\mathcal{D}}$ 2 % If timer is non-zero and no regions have been claimed since last update, perform trivial update $\begin{array}{l} \text{if } T_i > 0 \text{ and } P_i^* = P_i \text{ then} \\ | \quad \text{Set } \tau_i^A = \tau_i^A - t_0 + \omega_i \text{ and } \omega_i^A = \omega_i = t_0 \end{array}$ 3 else % Iterate through generator locations to find a cost-minimizing configuration for $k \in P_i^{I\mathcal{D}}$ do Set $P_i^{\text{test}} = P_i^{\text{add}}(k)$ and $c_i^{\text{test}} = k$ if $\mathcal{H}(c^{\text{test}}, P^{\text{test}}, t_0) < \mathcal{H}(c^*, P^*, t_0)$ then \lfloor Set $P^* = P^{\text{test}}, c^* = c^{\text{test}}$ 4 5 % Update timers and variables Set $P_i^{A,\mathrm{pd}} = P_i^* \setminus P_i^{I\mathcal{D}}$ 6 Call Alg. 2 and obtain output $\Phi_i^A, \omega, T, \tau_i^A$ 7 Set $P_i = P_i^A = P_i^*$, $c_i = c_i^A = c_i^*$, $\omega_i^A = \omega_i$ for $k \in P_i$ do Set $ID_k = i$ 8 9 return $P, c, P_i^A, c_i^A, P_i^{A, \text{pd}}, T, \tau_i^A, \Phi_i^A, \omega, \omega_i^A, I\mathcal{D}$ 10

the input $\Delta_H > 0$ is a constant parameter⁶. Recall from Section 3.3 that, given a vertex k and sets $Q'' \subset Q'$, the value $d_{Q'}(k,Q'')$ represents the minimum time required to travel from k to any node in the set Q''. Therefore, the operations in lines 1 and 2 of Algorithm 2 are implicitly max-min operations that, intuitively, define upper bounds on the time required for the agents to vacate areas that have shifted as a

result of the update. Additional remarks to aid in reader understanding are given by the comments within the algorithms (italicized tex preceded by a "%" character).

Algorithm 2: Timer Update		
Data : $t_0, P, P^*, c^*, P_i^{A, pd}, \Phi, \omega, \overline{\Delta}, \Delta_H, T, s$ Result : $\Phi_i^A, \omega, T, \tau_i^A$		
begin		
	% Find the worst-case time required for agent i to return to its remaining coverage region from any node that was removed during the update.	
1	$\Delta_i^{\mathrm{Bf}} \coloneqq \max\left\{ rac{1}{s_i} d_{P_i}\left(k, P_i^* ackslash P_i^{A, \mathrm{pd}} ight) \mid k \in P_i ackslash P_i^* ight\}$	
	% Find the worst-case time required for other	
	agents to vacate regions acquired by agent i, and	
	redefine timers to ensure communication	
	for Each $j \neq i$ satisfying $P_j \cap P_i^* \neq \emptyset$ do	
2	$ \left \begin{array}{c} \Delta_{j}^{\mathrm{Bf}} \coloneqq \max\left\{ \frac{1}{s_{j}} d_{P_{j}}\left(k, P_{j} \setminus P_{i}^{*}\right) \mid k \in P_{j} \cap P_{i}^{*} \right\} \right. $	
3		
	% Select maximum and redefine variables	
4	Find $\Delta_{\max}^{\mathrm{Bf}} = \max_{j \neq i, P_j \cap P_i^* \neq \emptyset} \{\omega_j + \overline{\Delta} + \Delta_j^{\mathrm{Bf}} - t_0\}$	
5	Redefine $\Delta_{\max}^{Bf} = \max{\{\Delta_{\max}^{Bf}, \Delta_i^{Bf}\}}$	
6	Set $T_i = \Delta_{\max}^{Bf} + \Delta_{H}$, $\tau_i^A = \Delta_{\max}^{Bf}$, $\omega_i = t_0$	
7	Construct Φ_i^A (Eq. (1)) with updated variables	
8	return $\Phi_i^A, \omega, T, \tau_i^A$	

Consider the following initialization assumptions.

Assumption 1 (Initialization). *The following properties are satisfied when* t = 0*:*

1. *P* is a connected *m*-partition of *Q*, 2. $P = P^{I\mathcal{D}}$, and 3. for all $i \in \{1, ..., m\}$, (a) $P_i = P_i^A$, (b) $c_i = c_i^A \in P_i^A$, (c) $P_i^{A, \text{pd}} = \emptyset$, (d) $T_i = \omega_i = \omega_i^A = 0$, and (e) $\tau_i^A = -\Delta_H$.

Notice 1 and 3b together imply that $c_i \neq c_j$ for any $j \neq i$. Our first result guarantees well-posedness of Algorithm 1.

Theorem 1 (Well-Posedness). Under Assumption 1, a scheme in which, during each agent-cloud exchange, the cloud executes Algorithm 1 to update relevant global and local variables is well-posed. That is, operations required by Algorithm 1 are well-posed at the time of execution.

Algorithm 1 does not ensure that coverage regions (elements of *P*) remain disjoint. It does, however, guarantee that the *m*-covering *P*, the local coverage regions $P^A := \{P_i^A\}_{i=1}^m$,

 $^{^{6}\}Delta_{H}$ represents, loosely, the amount of time an agent must hold a vertex before it can be reassigned. Precise characterization is in Section 4.3.

and the local likelihoods $\{\Phi_i^A\}_{i=1}^m$ retain properties that are consistent with a decomposition-based scheme. Namely, the coverings *P* and *P*^A maintain connectivity, and each Φ_i^A has support that is disjoint from that of all other local likelihoods, yet evolves to provide reasonable global coverage. Further, Algorithm 2 ensures that agents can "safely" vacate areas that are reassigned before newly assigned agents enter. We expand upon these ideas in the following two subsections.

4.3 Set Properties

The next result formalizes key set properties.

Theorem 2 (Set Properties). Suppose Assumption 1 holds, and that, during each agent-cloud exchange, the cloud executes Algorithm 1 to update relevant global and local variables. Then, the following hold at any time $t \ge 0$:

- 1. P^{ID} is a connected m-partition of Q,
- 2. P is a connected m-covering of Q,

3. $c_i \in P_i$ and $c_i \neq c_j$ for any $i \neq j$,

- 4. $\operatorname{supp}(\Phi_i^A(\cdot,t)) \subseteq P_i$ for any *i*, and
- 5. $\bigcap_{i=1}^{m} \operatorname{supp}(\Phi_{i}^{A}(\cdot,t)) = \emptyset.$

When the cloud makes additions to an agent's coverage region, newly added vertices are not immediately included in the instantaneous support of the agent's local likelihood. If agent movement is restricted to lie within this support, the delay temporarily prohibits exploration of newly added regions, allowing time for other agents to vacate. Conversely, when regions are removed from an agent's coverage region, Algorithm 1 ensures a "safe" path, i.e., a path with no collision risk, exists and persists long enough for the agent to vacate. Let $\overline{d} := \max_i \frac{1}{s_i} \sum_{\{k_1,k_2\} \in E} d_Q(k_1,k_2)$, and define agent *i*'s prohibited region, $\operatorname{Proh}_i(t)$, as the set of vertices not belonging to the support of $\Phi_i^A(\cdot,t)$, i.e., $\operatorname{Proh}_i(t) := \{k \in Q \mid k \notin \operatorname{supp}(\Phi_i^A(\cdot,t))\}$. We formalize this discussion here.

Theorem 3 (Coverage Quality). Suppose Assumption 1 holds, and that, during each agent-cloud exchange, the cloud updates relevant global and local coverage variables via Algorithm 1. Then, for any $k \in Q$ and any $t \ge 0$:

- 1. k belongs to at least one agent's coverage region P_i ,
- 2. *if* $k \in \text{Proh}_i(t) \cap P_i$ *for some i*, *then there exists* t_0 *satisfying* $t < t_0 < t + \overline{\Delta} + \overline{d}$ *such that, for all* $\overline{t} \in [t_0, t_0 + \Delta_H]$, *the vertex k belongs to the set* $P_i \setminus \text{Proh}_i(\overline{t})$, *and*
- 3. if k is removed from P_i at time t, then, for all

$$\bar{t} \in \left(t, t + \frac{1}{s_i} d_{P_i^-}\left(k, P_i^{I\mathcal{D}, -}\right)\right]$$

we have

- (a) $P_i^{I\mathcal{D},-} \subseteq P_i$, and
- (b) there exists a length-minimizing path on $G(P_i^-)$ from k into $P_i^{I\mathcal{D},-}$, and all of the vertices along any such path (except the terminal vertex) belong to the set $\operatorname{Proh}_{ID_k^+}(\overline{t}) \setminus \bigcup_{j \neq ID_k^+} P_j$.

Theorems 2 and 3 allow Algorithm 1 to operate within a decomposition-based framework to provide reasonable coverage with inherent collision avoidance. Indeed, if agents avoid prohibited regions, the theorems imply that each agent (i) can visit its entire coverage region (connectedness), (ii) allows adequate time for other agents to vacate reassigned regions, and (iii) has a "safe" route into the remaining coverage region if its current location is removed during an update.

Remark 3 (Coverage Variables). If Assumption 1 holds and updates are performed with Algorithm 1, then $P_i = P_i^A$ and $c_i = c_i^A$ for all i and all t. Thus, both Theorem 2 and Theorem 3 are equivalently stated by replacing P_i with P_i^A and c_i with c_i^A in their respective theorem statement.

Remark 4 (Bounds). Theorem 3 holds if \overline{d} is replaced by any upper bound on the distance between two arbitrary vertices within an arbitrary connected subgraph of *G*.

4.4 Convergence Properties

Our proposed strategy differs from [12] due to logic, i.e., timing parameters, etc., that ensures effective operation within a decomposition-based framework. Note also that \mathcal{H} differs from previous partitioning cost functions in [12, 26, 30], since it uses subgraph, rather than global graph, distances. As such, convergence properties of the algorithms herein do not follow readily from existing results. Consider the following definition.

Definition 2 (Pareto Optimality). *The* pair (c,P) is *Pareto optimal at time t if*

1.
$$\mathcal{H}(c,P,t) \leq \mathcal{H}(\bar{c},P,t)$$
 for any $\bar{c} \in Q^m$, and
2. $\mathcal{H}(c,P,t) \leq \mathcal{H}(c,\bar{P},t)$ for any $\bar{P} \in Cov_m(Q)$

When Φ is time-invariant (and Assumption 1 holds), Algorithm 1 produces finite-time convergence of coverage regions and generators to a Pareto optimal pair. The limiting coverage regions are "optimal" in that they balance the sensing load in a way that directly considers the event likelihood. Further, the operation only requires sporadic and unplanned agent-cloud exchanges. We formalize this result here.

Theorem 4 (Convergence). Suppose Assumption 1 holds and that, during each agent-cloud exchange, the cloud updates relevant global and local coverage variables via Algorithm 1. If Φ is time-invariant, i.e., $\Phi(\cdot,t_1) = \Phi(\cdot,t_2)$ for all t_1,t_2 , then the m-covering P and the generators c converge in finite time to an m-partition P^{*} and a set c^{*}, resp. The pair (c^{*}, P^{*}) is Pareto optimal at any time following convergence.

Remark 5 (Weighted Voronoi Partitions). It can be shown that Pareto optimality of (c^*, P^*) in Theorem 4 implies that, following convergence, P^* is a multiplicativelyweighted Voronoi partition (generated by c^* , weighted by s, subject to density $\Phi(\cdot,t)$) by standard definitions (e.g., [12]). If the centroid set of each P_i is defined as $\arg\min_{h \in P_i} \sum_{k \in P_i} d_{P_i}(k,h)\Phi(k,t)$, then P^* is also centroidal. Our coverage scheme balances the sensing load in that it updates coverage responsibilities in a way that locally minimizes the expected time required for an appropriate agent to travel from its region generator to a randomly occurring event within the environment. In essence, this serves to avoid unreasonable configurations, e.g., configurations where one agent is assigned responsibility of all important areas and remaining agents are only given non-important regions. Further, the update rules consider agent speeds, so faster agents will generally be assigned larger (weighted by the likelihood) coverage regions than slower agents. Similar strategies are employed in traditional load-balancing algorithms that are based on Voronoi partitions and operate over environments with stochastic event likelihoods, e.g., [12, 26, 30].

Theorem 4 and Remark 5 provide a rigorous characterization of the type of load balancing provided in the static likelihood case. A few comments are in order. First, Pareto optimal pairs (and multiplicatively-weighted, centroidal Voronoi partitions) are non-unique. Theorem 4 only guarantees that one possible Pareto optimal pair will be found in the static likelihood case, and does not exclude the existence of a lower-cost configuration. Second, the coverage configurations produced by our algorithms may not be *equitable*, i.e., the probability of events in each coverage region may vary across agents, even in the static case. The development of a strategy that produces equitable partitions within a similar cloud-supported architecture is an open problem and an interesting area of future research.

5 Decomposition-Based Surveillance.

This section pairs the proposed partitioning framework with a generic, single-vehicle trajectory planner, forming the complete, multi-agent surveillance framework.

5.1 Complete Routing Algorithm

By Theorem 2, the support of each Φ_i^A (i) lies entirely within the coverage region P_i^A , and (ii) is disjoint from the support of other local likelihoods. By Theorem 3, (i) any vertex can only go uncovered over bounded intervals, and (ii) the parameter Δ_H is a lower bound on the time that a recently uncovered vertex must remain covered before it can become uncovered again. These results suggest that an intelligent routing scheme that carefully restricts motion according to the instantaneous support of the local likelihood functions could achieve adequate coverage while maintaining collision avoidance. This motivates the following assumption.

Assumption 2 (Agent Motion). Each agent i has knowledge of its position at any time t, and its on-board trajectory planner operates under the following guidelines:

- 1. generated trajectories obey agent motion constraints,
- 2. trajectories are constructed incrementally and can be altered in real-time, and
- the agent is never directed to enter regions associated with Proh_i(t).

Each agent precisely traverses generated trajectories.

Note that condition 2 of Assumption 2 implies that the agent is never directed to leave regions associated with P_i^A . Algorithm 3 presents the local protocol for Agent *i*. Here, the on-board trajectory planner is used to continually update agent trajectories as the mission progresses (line 1). As such, the low-level characteristics of each individual agent's motion (i.e., the relation between the underlying likelihood function, the coverage configuration, and the resultant trajectory) depends on the particular planner employed.

Algorithm 3: Motion Protocol for Agent <i>i</i>		
Data : $G(O)$, Φ_i^A , P_i^A , c_i^A , $P_i^{A,pd}$, τ_i^A , ω_i^A		
begin		
while <i>True</i> do		
1	Utilize on-board planner to update trajectory	
2	Follow trajectory	
	if Data link with the cloud then	
3	Set $P_i^{\text{test}} = P_i^A$	
4	Obtain updated variables from the cloud	
	if Location lies within $P^{\text{test}} \setminus P^A$ then	
5	Find shortest path in $G(P^{\text{test}})$ from	
2	the currently occupied node into P^A	
	the currently occupied node into T_i	
	while Agent i is outside P_i^{i} do	
6	Follow the aforementioned route	

5.2 Collision Avoidance

Although Assumption 2 locally prevents agents from entering prohibited regions, dynamic coverage updates can still place an agent within its prohibited region if the vertex corresponding to its location is abruptly removed during an update. If this happens, Algorithm 3 constructs a route from the agent's location back into a region where there is no collision risk. With mild assumptions, this construction (i) is well-defined, and (ii) does not present a transient collision risk. We formalize this result here.

Theorem 5 (Collision Avoidance). Suppose Assumptions 1 and 2 hold, and that each agent's initial position lies within its initial coverage region P_i . If each agent's motion is locally governed according to Algorithm 3, where the update in line 4 is calculated by the cloud via Algorithm 1, then no two agents will ever collide.

Remark 6 (Agent Dynamics). We assume point mass dynamics for simplicity. However, all theorems herein also apply under alternative models, e.g., non-holonomic dynamics, provided that the surveillance environment is discretized so that (i) travel between adjacent grid elements is possible without leaving their union, (ii) agents can traverse the aforementioned paths at maximum speed, and (iii) edge weights accurately upper bound travel between adjacent regions. Typically, these conditions can be met by choosing discretization cells that are sufficiently large. For example, under a Dubins vehicle model, choosing square cells whose edge lengths are at least twice the minimum turning radius of any of the vehicles is sufficient. If these conditions are not met, Theorems 1, 2, 3, and 4 still apply, though Theorem 5 is no longer guaranteed since more sophisticated logic would be needed to ensure agent trajectories remain within the allowable regions. The development of an algorithmic extension that would guarantee collision avoidance for general non-holonomic vehicles is not straightforward and is left as a topic of future research.

In practice, however, we note that even if Theorem 5 is not satisfied in a strict sense, implementation of the algorithms herein within a decomposition-based scheme will usually still provide a significantly reduced collision or redundant sensing risk, provided that vehicles remain close to the allowable regions specified by the updates.

6 Numerical Examples

This section presents numerical examples to illustrate the proposed framework's utility. In all examples, updates are performed on the cloud via Algorithm 1 during each agent-cloud exchange, and each agent's local processor runs the motion protocol in Algorithm 3. For incremental trajectory construction (Algorithm 3, line 1), all examples use a modified Spectral Multiscale Coverage (SMC) scheme [10], which creates trajectories to mimic ergodic dynamics while also locally constraining motion to lie outside of prohibited regions. Notice this planner satisfies Assumption 2. Initial region generators were selected randomly (enforcing noncoincidence), and each agent was initially placed at its region generator. The initial covering P was created by calculating a weighted Voronoi partition, and remaining initial parameters were chosen to satisfy Assumption 1. It is assumed that relevant initial variables are uploaded to the agents' local servers prior to initial deployment, i.e., each agent has full knowledge of relevant initial information at time 0. For each simulation, randomly chosen agents sporadically exchanged data with the cloud. Agent-cloud exchange times were randomly chosen, subject to a maximum inter-exchange time $\overline{\Delta}$.

6.1 Time-Invariant Likelihood

Consider a 4 agent mission, executed over a 100 x 100 surveillance region that is subject to a time-invariant, Gaussian likelihood centered near the bottom left corner. The region is divided into 400, 5 x 5 subregions. Regions are considered adjacent if they share a horizontal or vertical edge. Here, each agent had a maximum speed of 1 unit distance per unit time, and $\overline{\Delta} = 10$ time units. Fig. 3 shows the evolution of the coverage regions for an example simulation run. Agent trajectories are shown by the colored lines. Note that Fig. 3 only shows each agent *i*'s *active* coverage region, i.e., $P_i \setminus \text{Proh}_i(t)$. The family of active coverage regions does not generally form an *m*-covering of Q; however, elements of this family are connected and never intersect as a result of inherent collision avoidance properties.



Fig. 3. A 4 agent example mission over a static Gaussian likelihood. Each agent's position, past trajectory, and active coverage region are shown with the colored triangle, line, and squares, resp.

The left plot in Fig. 4 depicts the maximum amount of time that any grid square went uncovered, i.e. the grid square did not belong to any agent's active covering, during each of 50 simulation runs. Here, the maximum amount of time that



Fig. 4. The maximum amount of time that any subregion went uncovered in each of 50 simulation runs (left), and the value of the cost \mathcal{H} as a function of time, averaged over the same 50 runs (right).

any region went uncovered was 186 units, though the maximum for most trials was less than 75 units. This is wellbelow the loose bound $\overline{\Delta} + \overline{d} = 770$ predicted by Theorem 3 (see Remark 4). Note that this metric does not capture the time between the agent's actual visits to the grid-square, only the length of intervals on which no agent was *allowed* to visit the square. The time between visits is governed by the particular choice of trajectory planner and the parameter $\Delta_{\rm H}$.

The right plot in Fig. 4 shows the mean values of the cost function \mathcal{H} , calculated over the same 50 simulations runs. Here, error bars represent the range of cost values at select points. The variance between runs is due to the stochastic nature of the agent-cloud exchange patterns. Notice the cost is non-increasing over time, eventually settling as the coverage regions/generators reach their limiting configuration, e.g., see Fig. 3. These configurations are each Pareto optimal and form a multiplicatively-weighted Voronoi partition (Remark 5). The resultant coverage assignments provide load-balancing that takes into account the event likelihood. If the low-level trajectory planner biases trajectories according to the event likelihood, this results in desirable coverage proper-

ties. Under the modified SMC planner used here, the temporal distribution of agent locations closely resembles the spatial likelihood distribution in the limit, as shown in Fig. 5.



Fig. 5. Comparison between the (time-invariant) event likelihood Φ (left), and the proportion of time that some agent occupied each subregion after significant time has passed (10000 units) (right).

Further, during the simulation, no two agents ever occupied the same space due to the careful parameter manipulations employed by Algorithm 1. Fig. 6 illustrates the logic governing these manipulations through a simplistic example: During the first update, the blue agent acquires some of the



Fig. 6. Simplified example illustrating how Algorithm 1 manipulates timing parameters to prevent agent collisions.

red agent's coverage region. Rather than immediately adding these regions to its active covering, the blue region waits until sufficient time has passed to guarantee that the red agent has updated and moved out of the reassigned regions. Under Algorithm 3, once the red agent communicates with the cloud, it immediately vacates the reassigned regions, after which the blue agent can add the region to its active covering. This procedure guarantees that no two agents will never have overlapping active coverings and thus never collide (Theorem 5). This same logic results in inherent collision prevention over more complex scenarios.

We can also compare the coverage regions produced by Algorithm 1 to those produced by the partitioning algorithm in [12]. The two algorithms were simulated in parallel, performing updates with the same randomly chosen agent-cloud exchange orderings across the two conditions. The left and the right plots in Fig. 7 show the mean coverage cost over 50 simulation runs, calculated using \mathcal{H}_{min} (defined in [12], Section II-C) and \mathcal{H} (Section 4.2), respectively (portions of the curves extending above the axes indicate an infinite value). The function \mathcal{H}_{min} is defined nearly identically to \mathcal{H} , but uses global graph distances, rather than subgraph distances. It is clear that the evolution produced by the algorithm in [12] converges to a final configuration slightly faster than that produced by Algorithm 1 whenever costs are quantified using \mathcal{H}_{min} . However, when costs are calculated using \mathcal{H} , the algorithms in [12] produced intermediate configurations with infinite cost, indicating disconnected regions, while Algorithm 1 maintained connectivity. In contrast to [12], our surveillance framework allows for complete coverage without requiring the agents to leave their assigned regions, allowing it to operate more effectively within a multi-agent surveillance scheme.



Fig. 7. Comparison of coverage cost between [12] and Algorithm 1. Coverage costs are calculated with \mathcal{H}_{min} ([12], Section II-C) on the left and with \mathcal{H} (Section 4.2) on right.

6.2 Time-Varying Likelihood

We now illustrate how the proposed coverage framework reacts to changes in the underlying likelihood. Specifically, we study a particular type of time-varying likelihood in which the spatial distribution only changes at discrete timepoints, i.e., $\Phi(k, \cdot)$ is piece-wise constant for any $k \in Q$. This type of scenario is common in realistic missions, e.g., when the cloud's estimate of the global likelihood is only re-formulated if some agent's sensor data indicates a drastic change in the underlying landscape. For this purpose, we adopt identical parameters as in the first example, with the exception of the likelihood Φ , whose spatial distribution abruptly switches at select time-points. If the switches are sufficiently spaced in comparison to the rate of convergence, then the coverage regions dynamically adjust to an optimal configuration that is reflective of the current state. For example, Fig. 9 shows the coverage region evolution after the underlying likelihood undergoes a single switch between the likelihoods in Fig. 8 at time t = 2000. In contrast, if the



Fig. 8. The initial and final likelihood $\Phi(\cdot, t)$.

likelihood changes faster than the rate of convergence, coverage regions are constantly in a transient state. Despite this,



Fig. 9. Coverage regions after the likelihood switches (see Fig. 8)

the proposed framework still provides some degree of loadbalancing. To illustrate, the left plot in Fig. 10 shows the value of \mathcal{H} during a simulation in which the underlying likelihood switches at 12 randomly chosen time-points over a 1000 unit horizon. Each switch re-defined the spatial likelihood as a Gaussian distribution centered at a randomly selected location. Notice that the cost monotonically decreases between the abrupt spikes caused by changes in the underlying likelihood. A convergent state is never reached; however, coverage regions quickly shift away from high-cost configurations, as seen in the right plot of Fig. 10, which shows the average percentage drop in the value of the cost \mathcal{H} as a function of the number of non-trivial updates, i.e., updates that did not execute of Algorithm 1, line 3, following an abrupt switch in the likelihood. The percentage drop is calculated with respect to the cost immediately following the most recent switch. During the first nontrivial update, the cost drops on average 21.8% of the initial post-switch value, indicating a quick shift away from high-cost configurations.



Fig. 10. Evolution of the cost \mathcal{H} using a piecewise-constant likelihood with 12 random switches (indicated by the stars)(left), and the average percent decrease in \mathcal{H} following each switch (right).

7 Conclusion

This work develops a cloud-supported, decompositionbased, coverage control framework for multi-agent surveillance. In particular, a dynamic partitioning strategy balances the surveillance load across available agents, requiring only sporadic and unplanned agent-cloud exchanges. The partitioning update algorithm also manages high-level logic parameters to guarantee that the resulting coverage assignments have geometric and temporal properties that are amenable for combination with generic single vehicle trajectory planners. In certain cases, the proposed algorithms produce a Pareto optimal configuration, while ensuring collision avoidance throughout.

Future work should further relax communication assumptions to reflect additional limitations, e.g., use of directional antennae for wireless transmission. Extensions to the proposed algorithms to incorporate explicit area constraints on coverage regions, as well as more general vehicle dynamics, should also be explored. Other areas of future research include the combination of peer-to-peer and cloudbased communication, performance comparisons between specific trajectory planners when used within our framework, e.g., those involving ergodic Markov chains, and further theoretical performance characterization.

Appendix: Proofs

Proposition 2 (Sets). Suppose Assumption 1 holds, and that, at the time of each exchange occurring prior to a fixed time $\overline{t} \ge 0$, required algorithmic constructions are well-posed so that the cloud can perform updates via Algorithm 1. Then, for any $k \in Q$ and any time $t \le \overline{t}$:

- 1. $k \in P_{ID_k}$,
- 2. k belongs to at most 2 elements of P,
- 3. *if* $T_{ID_k} = 0$, *then* $k \notin P_{\ell}$ *for any* $\ell \neq ID_k$, *and*
- 4. if $k \in P_j$, $j \neq ID_k$, then $P_j \cap P_\ell^{I\mathcal{D}} = \emptyset$ for $\ell \notin \{j, ID_k\}$

Proof. Fix $\overline{t} \ge 0$, $k \in Q$. When t = 0, $P = P^{I\mathcal{D}}$ is an *m*partition of Q, implying the proposition. Since k is not removed from P_{ID_k} or added to any P_i with $i \neq \text{ID}_k$ until its first *reassignment*, i.e., when ID_k is changed, the proposition holds for all t prior to the first reassignment. Suppose the proposition holds for all t prior to the p^{th} reassignment, which occurs at time t_0 . Suppose $ID_k^- = j$, $ID_k^+ = i \neq j$. Alg. 1 defines $P_i^{I\mathcal{D},+} = P_i^+ = P_{ID_k^+}^+$. Thus, $k \in P_{ID_k^+}^+ = P_i^+$ and remains in these sets until another reassignment. Thus, statement 1 holds for all t prior to the $p + 1^{st}$ reassignment. Now note that, by Alg. 2, reassignment cannot occur at t_0 unless $T_i^- = 0$. By inductive assumption, statement 3 of the proposition holds when $t = t_0^-$, implying $k \notin P_\ell^-$ for any $\ell \neq j$. Upon reassignment, the timers T_i, T_i are modified such that $T_j^+, T_i^+ > \omega_j^+ + \overline{\Delta} - t_0$. Since (i) ID_k cannot change when $T_j > 0$, and (ii) agent j exchanges data with the cloud and removes k from P_j prior to time $\omega_j^+ + \overline{\Delta}$, we deduce that k solely belongs to P_j , P_i until the $p + 1^{st}$ reassignment. Further, for any $t \ge t_0^+$ at which $T_i = 0$ and the $p + 1^{st}$ reassignment has not yet occurred, $k \in P_i$ exclusively (addition to other sets in *P* without reassignment is impossible). We deduce statements 2 and 3 for any *t* prior to the $p + 1^{st}$ reassignment. Finally, considering Alg. 2, it is straightforward to show that $T_j^- = 0$ implies $P_j^- = P_j^{I\mathcal{D},-}$ ($T_j = 0$ only if the most recent exchange that manipulated elements of $P_j^{I\mathcal{D}}$ involved agent *j*, after which $P_j = P_j^{I\mathcal{D}}$). Further, (i) no agent claims vertices from P_j^+ unless $T_j = 0$, and (ii) no vertex is added to a coverage region without reassignment. As such, $P_j \cap P_\ell^{I\mathcal{D}} = \emptyset$ for any $\ell \notin \{j, ID_k = i\}$ prior to another update in which some other agent claims vertices from P_j . Extending this logic and noting the bound Δ , we deduce the same result for any *t* prior to the $p + 1^{st}$ reassignment of *k*. Noting Δ once again, the proposition follows by induction.

Proof of Theorem 1. It suffices to show that Def. 1 is wellposed (Prop. 1) whenever additive sets are required. We proceed by induction. When t = 0, $P^{I\mathcal{D}} = P$ is a connected *m*partition of Q; thus, for any $i, P_i^{I\mathcal{D}} \cap (\bigcup_{j \neq i} P_j) = \emptyset$. The same holds prior to the first agent-cloud exchange, so the first call to Alg. 1 is well-posed. Now assume that, for all t prior to the p^{th} call to Alg. 1, (i) $P^{I\mathcal{D}}$ is a connected *m*-partition of *Q*, and (ii) if an exchange occurs that requires construction of P_i^{add} (assume this also applies to the impending p^{th} exchange), then $P_i^{I\mathcal{D}} \cap \left(\bigcup_{j \neq i} P_j\right) = \emptyset$ immediately prior to the exchange. This implies that Prop. 2 holds at any time t prior to the $p + 1^{st}$ exchange. Assume the p^{th} exchange occurs at time t_0 and involves agent *i*. Recall that $P^{I\mathcal{D},+}$ is an *m*-partition of Q. To show $P^{I\mathcal{D},+}$ is connected, first notice $P_i^{I\mathcal{D},+} = P_i^+$. Since either $P_i^+ = P_i^{\text{add}}(c_i^+)$ (connected by Def. 1) or $P_i^{I\mathcal{D},+} = P_i^{I\mathcal{D},+}$ (connected by assumption), connectivity of $P_i^{I\mathcal{D},+}$ follows. Now consider $P_j^{I\mathcal{D}}$, $j \neq i$. If $T_j^- \neq 0$, then $P_j^{I\mathcal{D},+} = P_j^{I\mathcal{D},-}$ is connected. Suppose $T_j^- = 0$ and $P_j^{I\mathcal{D},+}$ is not connected. By Prop. 2, $P_j^+ \cap P_\ell^{I\mathcal{D},+} = \emptyset$ for any $\ell \notin \{i, j\}$. Thus, there exists $k_1 \in P_j^{I\mathcal{D},+}$ such that (i) $k_1 \notin P_i^{\text{add}}(c_i^+)$, and (ii) any optimal path in $G(P_i^+)$ spanning k_1 and c_i^+ contains some $k_2 \in P_i^{\text{add}}(c_i^+) = P_i^+$. Select one such path and vertex k_2 . Without loss of generality, assume $\{k_1, k_2\} \in E$. Def. 1 implies $\frac{1}{s_i} d_{P_i^+}(k_2, c_i^+) < \min\{\frac{1}{s_\ell} d_{P_\ell^+}(k_2, c_\ell^+) | \ell \neq i, k_2 \in I$ P_{ℓ}^+ and thus $\frac{1}{s_i}d_{P_i^+\cup\{k_1\}}(k_1,c_i^+) < \frac{1}{s_j}d_{P_j^+}(k_1,c_j^+)$. Since
$$\begin{split} T_{j}^{-} &= 0 \text{ and } \mathrm{ID}_{k_{1}}^{-} = j, \text{ Prop. 2 implies } \frac{1}{s_{i}} d_{P_{i}^{+} \cup \{k_{1}\}}(k_{1}, c_{i}^{+}) < \\ \frac{1}{s_{j}} d_{P_{j}^{+}}(k_{1}, c_{j}^{+}) &= \min\{\frac{1}{s_{\ell}} d_{P_{\ell}^{+}}(k_{1}, c_{\ell}^{+}) \,|\, \ell \neq i, k_{1} \in P_{\ell}^{+}\}, \text{ contra-} \end{split}$$
dicting $k_1 \notin P_i^{\text{add}}(c_i^+)$. Thus, $P_i^{I\mathcal{D},+}$ is connected. Invoking Prop. 2 statement 3, the inductive assumption holds for all tprior to the $p + 1^{st}$ exchange, implying well-posedness.

Proof of Theorem 2.

<u>Statement 1</u>: The proof of Thm. 1 implies the statement. <u>Statement 2</u>: *P* is an *m*-covering of *Q* since $P^{I\mathcal{D}}$ is an *m*-partition of *Q*, and $P_i^{I\mathcal{D}} \subseteq P_i$ for any *i* (Prop. 2, statement 1). The covering *P* is connected, since $P_i = P_i^{I\mathcal{D}}$ (connected by statement 1) immediately following any agent-cloud exchange and is unchanged in between updates.

<u>Statement 3</u>: It suffices to show $ID_{c_i} = i$ for any t, i: this

would imply $c_i \neq c_j$ for any $i \neq j$, and $c_i \in P_i$ (Prop. 2). Since $ID_{c_i} = i$ for all i at t = 0, the same holds for any t prior to the first agent-cloud exchange. Suppose $ID_{c_i} = i$ for all i (thus $c_i \neq c_j$ for any $i \neq j$) prior to the p^{th} exchange. If agent i is the p^{th} communicating agent, lines 2, 9 of Alg. 1 imply $ID_{c_i^+}^+ = i$. Since $d_{P_j^-}(c_j^-, c_j^-) = 0$ for any j, we have $c_j^+ \notin P_i^{\text{add}}(c_i^+)$. Thus, $ID_{c_i^+}^+ = j$, and induction proves the statement.

Statements 4 and 5: Statement 4 follows from (1), noting that $P_i^A = P_i$. Statement 5 holds by assumption when t = 0. Let $k \in Q$, and consider times when ID_k changes (k is reas*signed*). Since supp $(\Phi_j^A(\cdot, t)) = P_j = P_j^A$ for any j at t = 0, statement 4 implies that, for any t prior to the first reassignment, k belongs exclusively to $supp(\Phi^A_{ID_k}(\cdot,t))$. Suppose statement 5 holds for all t prior to the p^{th} reassignment (occurring at time t_0), and $ID_k^- = j$, $ID_k^+ = i \neq j$. Then, $T_j^- = 0$ and k belongs exclusively to P_j^- when $t = t_0^-$ (Prop. 2). By Alg. 1 and 2, $k \in P_i^{A, \text{pd}, +}$ and $T_i^+ > \omega_j^+ + \overline{\Delta} - t_0 \ge \tau_i^{A, +}$. Since $\operatorname{supp}(\Phi_i^A(\cdot,t))$ is unchanging over an interval of length at least $T_i^+ \ge \tau_i^{A,+}$, (1) implies $k \notin \operatorname{supp}(\Phi_i^A(\cdot,t))$ when $t \in [t_0^+, t_0^+ + \tau_i^{A,+}]$. Since k is reassigned when $t = t_0, k \in$ $P_i^+ \setminus P_i^{I\mathcal{D},-}$ and $T_i^+ = \omega_i^+ + \overline{\Delta} - t_0$. Agent *j* will communicate with the cloud at some time $t_1 < t_0 + T_i^+ = \omega_i^+ + \overline{\Delta} < t_0 + T_i^+$. Thus, $T_i > 0$ when $t = t_1$, and k is removed from both P_j and $\operatorname{supp}(\Phi_i^A(\cdot,t))$. Thus, for all $t > t_0 + \tau_i^+$ and before the $p + 1^{\operatorname{st}}$ reassignment, k belongs exclusively to $supp(\phi_i(\cdot, t))$.

<u>Statement 2</u>: For any *i*, (i) $T_i = 0$ when t = 0, and (ii) $\frac{1}{s_i} d_{Q'}(k_1, k_2) \leq \overline{d}$ for any connected $Q' \subseteq Q$, $k_1, k_2 \in Q'$. Thus, it is straightforward to show, for any *i*, *t*, we have the bound $T_i \leq \overline{\Delta} + \Delta_{\rm H} + \overline{d}$. We show by induction that, for any *i*, *t*, the bound $\tau_i^A - t + \omega_i^A \leq T_i - \Delta_{\rm H}$ also holds: $T_i = 0$ and $\tau_i^A = -\Delta_{\rm H}$ when t = 0, so $\tau_i^A - t + \omega_i^A = \tau_i^A \leq T_i - \Delta_{\rm H}$, and the bound holds prior to the first cloud-agent exchange involving *any* agent, since $\tau_i^A - t = \tau_i^A - t + \omega_i^A \leq -\Delta_{\rm H} \leq T_i - \Delta_{\rm H}$ at any such time. Assume the bound holds prior to the *p*th exchange (occuring at $t = t_0$). Consider 2 cases: if agent *i* is the communicating agent, then $\tau_i^{A,+} - t + \omega_i^A = \tau_i^+ := T_i^+ - \Delta_{\rm H}$; if not, then $\tau_i^{A,+} = \tau_i^{A,-}$ and either (i) $T_i^- = T_i^+$ implying the desired bound, or (ii) $T_i^- = 0$ and $\tau_i^{A,+} - t_0 + \omega_i^{A,+} = \tau_i^{A,-} - t + \omega_i^{A,-} \leq T_i^- - \Delta_{\rm H} = -\Delta_{\rm H} \leq (\omega_i^{A,+} + \overline{\Delta} - t_0) - \Delta_{\rm H} = T_i^+ - \Delta_{\rm H}$. This logic extends to all *t* prior to the *p* + 1st exchange and the desired bound follows by induction.

Using the aforementioned two bounds, we have $\tau_i^A + \omega_i^A \leq t + \overline{\Delta} + \overline{d}$. Fix t and $k \in \operatorname{Proh}_i(t) \cap P_i$. Then, $k \in P_i^{A,+} = P_i^+$, $k \in P_i^{A,\operatorname{pd},+}$, and $t - \omega_i^{A,+} < \tau_i^{A,+}$ ('+' is with respect to the fixed time t). Further, over the interval $[t, \omega_i^{A,+} + \tau_i^{A,+}]$, the vertex k is not reassigned, P_i is not augmented, and τ_i^A is unchanged. Therefore, $k \notin \operatorname{Proh}_i(\omega_i^{A,+} + \tau_i^{A,+})$. Setting $t_0 \coloneqq \omega_i^{A,+} + \tau_i^{A,+}$, we have $t < t_0 \leq t + \overline{\Delta} + \overline{d}$. Since $T_i \geq \tau_i^A + \Delta_H$ at time $\omega_i^{A,+}$, k is not reassigned during the interval $[\omega_i^{A,+}, \omega_i^{A,+} + T_i^{+}] \supseteq [\omega_i^{A,+}, t_0 + \Delta_H] \supseteq [t_0, t_0 + \Delta_H]$. Thus $k \in P_i \setminus \operatorname{Proh}_i(\cdot)$ over the same time interval.

<u>Statement 3</u>: Fix t and suppose $k \in P_i^- \setminus P_i^+$ (in this proof,

'+,-' are with respect to *t*). Then, (i) ID_k changed (*k* was reassigned) at time $t_0 < t$, (ii) agent *i* exchanges data with the cloud at time *t*, and (iii) no exchanges involving agent *i* occurred during the interval $[t_0,t)$. Upon reassignment at time t_0 , Alg. 2 specifies that (i) T_i is reset to value $\omega_i^{A,-} + \overline{\Delta} - t_0$, thus $P_i^{I\mathcal{D}}$ is unchanged over the interval $[t_0,t)$, (ii) *k* is added to $P_{\text{ID}_k}^{A,\text{pd}}$, and (iii) $\tau_{\text{ID}_k}^A$, T_{ID_k} are given values of at least

$$\tilde{\boldsymbol{\omega}} \coloneqq \max_{\tilde{k} \in P_i^- \setminus P_i^+} \left\{ \boldsymbol{\omega}_i^{A,-} + \overline{\Delta} + \frac{1}{s_i} d_{P_i^-} \left(\tilde{k}, P_i^{I\mathcal{D},-} \right) - t_0 \right\},\$$

implying that P_{ID_k} , $\operatorname{Proh}_{\text{ID}_k}(\cdot)$ are constant over the interval $(t_0, \tilde{\omega}] \supseteq (t_0, t + \frac{1}{s_i} d_{P_i^-}(k, P_i^{I\mathcal{D}, -})] \supseteq (t_0, t].$

Since coverage regions are connected and non-empty (Thm. 2), and $P_i^- \cap P_\ell^{I\mathcal{D}} = \emptyset$ for any $\ell \notin \{i, \mathrm{ID}_k^+\}$ on the interval $(t_0, t]$ (Prop. 2), (i) there exists a path of length $d_{P_i^-}(k, P_i^{I\mathcal{D}, -})$ from k into $P_i^{I\mathcal{D}, -}$ and every vertex along any such path (except the terminal vertex) lies within $P_i^- \setminus P_i^+$, and (ii) $P_i^- \setminus P_i^+ \subseteq \operatorname{Proh}_{\mathrm{ID}_k^+}(\cdot)$ over the interval $(t_0, \tilde{\omega}] \supseteq [t, t + \frac{1}{s_i}d_{P_i^-}(k, P_i^{I\mathcal{D}, -})]$. Since (i) each vertex belongs to at most two coverage regions (Prop. 2), (ii) $k \in P_i^- \setminus P_i^+$, and (iii) no agent claims vertices within $\operatorname{Proh}_{\mathrm{ID}_k^+}(\cdot) \cap P_{\mathrm{ID}_k^+}$ when $T_{\mathrm{ID}_k^+} > 0$, vertices along the path (excluding the terminal vertex) do not belong to P_j with $j \neq \mathrm{ID}_k$ over $[t, t + \frac{1}{s_i}d_{P_i^-}(k, P_i^{I\mathcal{D}, -})]$. To complete the proof, Alg. 2 implies $T_i^+ > \frac{1}{s_i}d_{P_i^-}(k, P_i^{I\mathcal{D}, -})$, and thus $P_i^{I\mathcal{D}, -} \subseteq P_i^{I\mathcal{D}}$ over $[t, t + \frac{1}{s_i}d_{P_i^-}(k, P_i^{I\mathcal{D}, -})]$.

Proposition 3 (Cost). Suppose Assumption 1 holds and that, during each agent-cloud exchange, the cloud updates relevant global and local coverage variables via Alg. 1. If $\Phi(\cdot,t_1) = \Phi(\cdot,t_2)$ for all t_1,t_2 , then $\mathcal{H}(c,P^{I\mathcal{D}},\cdot) = \mathcal{H}(c,P,\cdot)$.

Proof. Since Φ is time-invariant, $\mathcal{H}(\cdot, \cdot, t_1) = \mathcal{H}(\cdot, \cdot, t_2)$ for any t_1, t_2 . When t = 0, $P = P^{I\mathcal{D}}$ and $\mathcal{H}(c, P^{I\mathcal{D}}, 0) =$ $\mathcal{H}(c, P, 0)$. The same is true prior to the first agent-cloud exchange. Suppose that, prior to the p^{th} exchange (occurring at $t = t_0$, involving agent *i*), we have $\mathcal{H}(c^-, P^{I\mathcal{D},-}, t_0) =$ $\mathcal{H}(c^{-}, P^{-}, t_0)$. Recall that, for any j, P_i and $P_i^{I\mathcal{D}}$ coincide immediately following any exchange involving agent j and, if agent j claims vertices from P_i , then Alg. 2 ensures that agent *i* exchanges data with the cloud before additional vertices are claimed by other agents. Considering the pth update, this logic, along with Prop. 2, implies that $P_i^{I\mathcal{D},-} \cap P_j^- = \emptyset$, for all $j \neq i$. Noting that $c_i^+ \in P_i^{I\mathcal{D},-}$, we deduce that any $k \in P_i^{I\mathcal{D},-}$ contributes equivalently to $\mathcal{H}(c^+, P^{I\mathcal{D},+}, t_0)$ and $\mathcal{H}(c^+, P^+, t_0)$. If $k \in$ $P_i^{\text{add}}(c_i^+) \setminus P_i^{I\mathcal{D},-}$, then for any $j \neq i$ such that $k \in P_i^+$, we have $\frac{1}{s_i} d_{p_i^+}(k, c_i^+) < \frac{1}{s_i} d_{p_i^+}(k, c_i^+)$ (Def. 1), implying k contributes equivalently to $\mathcal{H}(c^+, P^{I\mathcal{D},+}, t_0)$ and $\mathcal{H}(c^+, P^+, t_0)$. Now suppose $k \in P_i^+ \setminus P_i^+$, where $P_i^+ \cap P_i^+ \neq \emptyset$. We show that $d_{P_i^{I\mathcal{D},+}}(c_j^+,k) = d_{P_j^+}(c_j^+,k)$: if a length-minimizing path

in $G(P_j^+)$ between c_j^+ and k is also contained in $G(P_j^{I\mathcal{D},+})$, the result is trivial. Suppose that every such minimum length path leaves $G(P_j^{I\mathcal{D},+})$. By Prop. 2, every $\bar{k} \in P_j^+$ must satisfy $\mathrm{ID}_{\bar{k}}^+ \in \{i, j\}$. Thus, assume without loss of generality that k is adjacent to P_i^+ . Let $\bar{k} \in P_i^+$ be a vertex that is adjacent to k and lies along a minimum-length path in $G(P_j^+)$ spanning c_j^+ and k. Since $\bar{k} \in P_i^+ \setminus P_i^{I\mathcal{D},-}$, we have $\bar{k} \in P_i^{\mathrm{add}}(c_i^+)$ as constructed during the update, implying $\frac{1}{s_i}d_{P_i^+}(\bar{k},c_i^+) < \min\{\frac{1}{s_\ell}d_{P_\ell^+}(\bar{k},c_\ell^+) \mid \ell \neq i, \bar{k} \in P_\ell^+\}$ and thus $\frac{1}{s_i}d_{P_i^+\cup\{k\}}(k,c_i^+) < \frac{1}{s_j}d_{P_j^+}(k,c_j^+)$. Since $T_j^- = 0$ and $\mathrm{ID}_k^- = j$, Prop. 2 implies $\frac{1}{s_i}d_{P_i^+\cup\{k\}}(k,c_i^+) < \frac{1}{s_j}d_{P_j^+}(k,c_j^+) = \min\{\frac{1}{s_\ell}d_{P_\ell^+}(k,c_\ell^+) \mid \ell \neq i, P_\ell^+\}$, contradicting $k \notin P_i^{\mathrm{add}}(c_i^+) \subset P_i^+$. Thus, $d_{P_j^{I\mathcal{D},+}}(c_j^+,k) = d_{P_j^+}(c_j^+,k)$, which, by inductive assumption, implies that k contributes equally to the value of both $\mathcal{H}(c^+, P^{I\mathcal{D},+}, t_0) = \mathcal{H}(c^+, P^+, t_0)$. Since $P, P^{I\mathcal{D}}$, and c are static between updates, the statement follows by induction.

Proof of Theorem 4. The cost $\mathcal{H}(c,P,t)$ is static in between agent-cloud exchanges, as P and c are unchanged. Consider an exchange occurring at time t_0 involving agent i. By Prop. 3, we have $\mathcal{H}(c^+, P^+, t_0) \leq \mathcal{H}(c^-, P^{I\mathcal{D}, -}, t_0) =$ $\mathcal{H}(c^-, P^-, t_0)$. Thus, the cost $\mathcal{H}(c,P,t)$ is non-increasing in time. Since $\operatorname{Cov}_m(Q)$ is finite, there is some time t_0 after which the value of \mathcal{H} is unchanging. Consider fixed $t > t_0$ at which some agent i exchanges data with the cloud. Since the value of \mathcal{H} is unchanging, Alg. 1 implies that $P^{I\mathcal{D}}$ and c are unchanged by the update. It follows that c and $P^{I\mathcal{D}}$ converge in finite time. Further, since $P_i^{I\mathcal{D},-} \cup P_i^{\operatorname{add}}(c_i^+) = P_i^+$. By persistence of exchanges imposed by $\overline{\Delta}$, this implies that after some finite time, P and $P^{I\mathcal{D}}$ are concurrent.

To prove Pareto optimality of the limiting configuration, consider t_0 , such that for all $t > t_0$, (c, P) is unchanging and P is an *m*-partition of Q. Timers are only reset when P is altered, so assume without loss of generality that $T_i = 0$ for all i at any $t > t_0$. Suppose agent i exchanges data with the cloud at time $t > t_0$. Alg. 1 implies that there is no $k \in P_i$ such that $\sum_{h \in P_i} d_{P_i}(h,k) \Phi(h,t) < \sum_{h \in P_i} d_{P_i}(h,c_i) \Phi(h,t)$ (if not, the cost is lowered by moving c_i). Similarly, for $k \in P_j$ with $j \neq i$ that is adjacent to P_i , we have $\frac{1}{s_i} d_{P_i \cup \{k\}}(c_i,k) \ge \frac{1}{s_j} d_{P_j}(c_j,k)$ (if not, there exists $k \in P_i^{\text{add}}(c_i^+) \setminus P_i^-$, contradicting convergence). As such, for any i, there is no $Q' \subset Q \setminus P_i$ such that $\sum_{k \in Q'} \frac{1}{s_i} d_{P_i \cup Q'}(c_i,k) < \sum_{k \in Q'} \min\{\frac{1}{s_j} d_{P_j}(c_j,k) | k \in P_j, j \neq i\}$, implying statement 2 of Def. 2.

Proof of Theorem 5. By Assumption 2, each agent's local trajectory planner never directs the agent into its prohibited region, so if no agent-cloud exchange occurs that removes the vertex corresponding to the relevant agent's location from its coverage region, then the statement is immediate. Suppose, at some time t_0 , agent *i*, whose location is associated with some $k \in P_i^{A,-}$, exchanges data with the cloud and *k* is removed, i.e., $k \notin P_i^{A,+}$. At time t_0^+ , agent *i* executes lines 5 and 6 of Alg. 3. Thm. 3 ensures that (i) there

exists a path in $G(P_i^{A,-})$ between k and $P_i^{I\mathcal{D},-}$, (ii) all vertices along the path belong to $\operatorname{Proh}_{\operatorname{ID}_k^+}(\bar{t}) \setminus \bigcup_{j \neq \operatorname{ID}_k^+} P_j^+$ whenever $\bar{t} \in (t_0, t_0 + \frac{1}{s_i} d_{P_i^{A,-}}(k, P_i^{I\mathcal{D},-})]$, and (iii) $P_i^{I\mathcal{D},-} \subseteq P_i \coloneqq P_i^A$ over the same interval. Thus, if agent i immediately moves along the path, it will lie exclusively within $\operatorname{Proh}_{\operatorname{ID}_k^+}(\cdot)$ until it reaches $P_i^{A,+}$. It remains to show that the agent does not enter $\operatorname{Proh}_i(\cdot) \cap P_i^+$ while traversing the aforementioned path. Without loss of generality, consider the update at time t_0 previously described. Since k is reassigned prior to the update, we have $\operatorname{Proh}_i(t_0^-) \cap P_i^- = \emptyset$ (vertices in P_i are not claimed unless $T_i = 0$, implying $t_0 - \omega_i^{A,-} > \tau_i^{A,-}$). By Prop. 2, we deduce $T_{\operatorname{ID}_k^+}^{+} > 0$, so no vertices in $P_i^{A,-} \cap P_{\operatorname{ID}_k^+}^{+}$ can belong to $\operatorname{Proh}_i(t_0^+)$. Since $T_i^+ > \tau_i^{A,+} > \frac{1}{s_i} d_{P_i^{A,-}}(k, P_i^{I\mathcal{D},-})$, $\operatorname{Proh}_i(\cdot)$ is constant over the interval $(t_0, t_0 + \frac{1}{s_i} d_{P_i^{A,-}}(k, P_i^{I\mathcal{D},-})]$.

References

- Macwan, A., Nejat, G., and Benhabib, B., 2011. "Target-motion prediction for robotic search and rescue in wilderness environments". *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, **41**(5), pp. 1287–1298.
- [2] Smith, R. N., Chao, Y., Li, P. P., Caron, D. A., Jones, B. H., and Sukhatme, G. S., 2010. "Planning and implementing trajectories for autonomous underwater vehicles to track evolving ocean processes based on predictions from a regional ocean model". *International Journal of Robotics Research*, **29**(12), pp. 1475–1497.
- [3] Dixon, S. R., and Wickens, C. D., 2006. "Automation reliability in unmanned aerial vehicle control: A reliance-compliance model of automation dependence in high workload". *Human Factors: The Journal of the Human Factors and Ergonomics Society*, **48**(3), pp. 474–486.
- [4] Pereira, A., Heidarsson, H., Oberg, C., Caron, D., Jones, B., and Sukhatme, G., 2010. "A communication framework for cost-effective operation of AUVs in coastal regions". In *Field and Service Robotics*, A. Howard, K. Iagnemma, and A. Kelly, eds., Vol. 62 of *Tracts in Advanced Robotics*. Springer, pp. 433–442.
- [5] Shah, R. C., Roy, S., Jain, S., and Brunette, W., 2003. "Data MULEs: modeling and analysis of a three-tier architecture for sparse sensor networks". *Ad Hoc Networks*, 1(2-3), pp. 215–233.
- [6] Peters, J., Srivastava, V., Taylor, G., Surana, A., Eckstein, M. P., and Bullo, F., 2015. "Mixed human-robot team surveillance: Integrating cognitive modeling with engineering design". *IEEE Control Systems*, 35(6), pp. 57–80.
- [7] Pasqualetti, F., Durham, J. W., and Bullo, F., 2012.
 "Cooperative patrolling via weighted tours: Performance analysis and distributed algorithms". *IEEE Transactions on Robotics*, 28(5), pp. 1181–1188.

- [8] Srivastava, V., Pasqualetti, F., and Bullo, F., 2013. "Stochastic surveillance strategies for spatial quickest detection". *International Journal of Robotics Research*, **32**(12), pp. 1438–1458.
- [9] Mathew, N., Smith, S. L., and Waslander, S. L., 2015.
 "Multirobot rendezvous planning for recharging in persistent tasks". *IEEE Transactions on Robotics*, **31**(1), pp. 128–142.
- [10] Mathew, G., and Mezić, I., 2011. "Metrics for ergodicity and design of ergodic dynamics for multi-agent systems". *Physica D: Nonlinear Phenomena*, 240(4), pp. 432–442.
- [11] Nigam, N., 2014. "The multiple unmanned air vehicle persistent surveillance problem: A review". *Machines*, 2(1), pp. 13–72.
- [12] Patel, R., Frasca, P., Durham, J. W., Carli, R., and Bullo, F., 2016. "Dynamic partitioning and coverage control with asynchronous one-to-base-station communication". *IEEE Transactions on Control of Network Systems*, 3(1), pp. 24–33.
- [13] Peng, J., and Akella, S., 2005. "Coordinating multiple robots with kinodynamic constraints along specified paths". *International Journal of Robotics Research*, 24(4), pp. 295–310.
- [14] Gerkey, B. P., and Matarić, M. J., 2002. "Sold!: Auction methods for multirobot coordination". *IEEE Transactions on Robotics and Automation*, 18(5), pp. 758–768.
- [15] Bonabeau, E., Dorigo, M., and Theraulaz, G., 1999. Swarm Intelligence: From Natural to Artificial Systems. No. 1 in Santa Fe Institute Studies in the Sciences of Complexity. Oxford university press.
- [16] Reif, J. H., and Wang, H., 1999. "Social potential fields: A distributed behavioral control for autonomous robots". *Robotics & Autonomous Systems*, 27(3), pp. 171–194.
- [17] Kariotoglou, N., Raimondo, D., Summers, S. J., and Lygeros, J., 2015. "Multi-agent autonomous surveillance: A framework based on stochastic reachability and hierarchical task allocation". ASME Journal of Dynamic Systems, Measurement, and Control, 137(3), p. 031008.
- [18] Shima, T., and Pagilla, P. R., 2007. "Special issue on analysis and control of multi-agent dynamic systems". *ASME Journal of Dynamic Systems, Measurement, and Control*, **129**(5), pp. 569–570.
- [19] Almeida, A., Ramalho, G., Santana, H., Tedesco, P., Menezes, T., Corruble, V., and Chevaleyre, Y., 2004.
 "Recent advances on multi-agent patrolling". In Advances in Artificial Intelligence, Vol. 3171 of Lecture Notes in Computer Science. Springer, pp. 474–483.
- [20] Pasqualetti, F., Zanella, F., Peters, J. R., Spindler, M., Carli, R., and Bullo, F., 2014. "Camera network coordination for intruder detection". *IEEE Transactions on Control Systems Technology*, **22**(5), pp. 1669–1683.
- [21] Latombe, J.-C., 1991. *Robot Motion Planning*. Kluwer Academic Publishers.
- [22] Bullo, F., and Smith, S. L., 2015. Lectures on Robotic

Planning and Kinematics. SIAM, Nov. To appear.

- [23] Yan, Z., Jouandeau, N., and Cherif, A., 2013. "A survey and analysis of multi-robot coordination". *International Journal of Advanced Robotic Systems*, **10**.
- [24] Bullo, F., Cortés, J., and Martínez, S., 2009. Distributed Control of Robotic Networks. Princeton University Press.
- [25] Okabe, A., Boots, B., Sugihara, K., and Chiu, S. N., 2000. Spatial Tessellations: Concepts and Applications of Voronoi Diagrams, 2 ed. Wiley Series in Probability and Statistics. John Wiley & Sons.
- [26] Cortés, J., Martínez, S., Karatas, T., and Bullo, F., 2004. "Coverage control for mobile sensing networks". *IEEE Transactions on Robotics and Automation*, **20**(2), pp. 243–255.
- [27] Patel, R., Frasca, P., and Bullo, F., 2014. "Centroidal area-constrained partitioning for robotic networks". ASME Journal of Dynamic Systems, Measurement, and Control, 136(3), p. 031024.
- [28] Cortés, J., 2010. "Coverage optimization and spatial load balancing by robotic sensor networks". *IEEE Transactions on Automatic Control*, **55**(3), pp. 749– 754.
- [29] Fjällström, P. O., 1998. "Algorithms for graph partitioning: A survey". *Linköping Electronic Articles in Computer and Information Science*, 3(10).
- [30] Durham, J. W., Carli, R., Frasca, P., and Bullo, F., 2012.
 "Discrete partitioning and coverage control for gossiping robots". *IEEE Transactions on Robotics*, 28(2), pp. 364–378.
- [31] Laporte, G., 1992. "The vehicle routing problem: An overview of exact and approximate algorithms". *European Journal of Operational Research*, **59**(3), pp. 345–358.
- [32] Toth, P., and Vigo, D., eds., 2001. *The Vehicle Routing Problem.* Monographs on Discrete Mathematics and Applications. SIAM.
- [33] Gutin, G., and Punnen, A. P., 2007. *The Traveling Salesman Problem and Its Variations*. Springer.
- [34] Ousingsawat, J., and Earl, M. G., 2007. "Modified lawn-mower search pattern for areas comprised of weighted regions". In American Control Conference, pp. 918–923.
- [35] Soltero, D. E., Schwager, M., and Rus, D., 2012. "Generating informative paths for persistent sensing in unknown environments". In IEEE/RSJ Int. Conf. on Intelligent Robots & Systems, pp. 2172–2179.
- [36] Lan, X., and Schwager, M., 2013. "Planning periodic persistent monitoring trajectories for sensing robots in Gaussian random fields". In IEEE Int. Conf. on Robotics and Automation, pp. 2415–2420.
- [37] Araujo, J. F., Sujit, P. B., and Sousa, J. B., 2013. "Multiple UAV area decomposition and coverage". In IEEE Symposium on Computational Intelligence for Security and Defense Applications, pp. 30–37.
- [38] Nigam, N., and Kroo, I., 2008. "Persistent surveillance using multiple unmanned air vehicles". In IEEE Aerospace Conference, pp. 1–14.

- [39] Maza, I., and Ollero, A., 2007. "Multiple UAV cooperative searching operation using polygon area decomposition and efficient coverage algorithms". In *Distributed Autonomous Robotic Systems 6*, R. Alami, R. Chatila, and H. Asama, eds. Springer, pp. 221–230.
- [40] Wood, J., and Hedrick, J., 2012. "Partition learning for multiagent planning". *Journal of Robotics*, 2012. Article ID 590479, 14 pages.
- [41] Bethke, B., Valenti, M., and How, J. P., 2008. "UAV task assignment". *IEEE Robotics & Automation Magazine*, **15**(1), pp. 39–44.
- [42] Kehoe, B., Patil, S., Abbeel, P., and Goldberg, K., 2015.
 "A survey of research on cloud robotics and automation". *IEEE Transactions on Automation Science and Engineering*, **12**(2), pp. 398–409.
- [43] Hale, M. T., and Egerstedt, M., 2015. "Differentially private cloud-based multi-agent optimization with constraints". In American Control Conference, pp. 1235– 1240.
- [44] Adaldo, A., Liuzza, D., Dimarogonas, D. V., and Johansson, K. H., 2015. "Control of multi-agent systems with event-triggered cloud access". In European Control Conference, pp. 954–961.