

Dynamic Partitioning and Coverage Control With Asynchronous One-to-Base-Station Communication

Rushabh Patel, *Student Member, IEEE*, Paolo Frasca, Joseph W. Durham, Ruggero Carli, and Francesco Bullo, *Fellow, IEEE*

Abstract—We propose algorithms to automatically deploy a group of mobile robots and provide coverage of a nonconvex environment with communication limitations. In settings, such as hilly terrain or for underwater ocean gliders, peer-to-peer communication can be impossible and frequent communication to a central base station may be impractical. This paper instead explores how to perform coverage control when each robot has only asynchronous and sporadic communication with a base station. The proposed algorithms rely upon overlapping territories, monotonically minimize suitable cost functions, and provably converge to a centroidal Voronoi partition. We also describe how the use of overlapping territories allows our algorithms to smoothly handle dynamic changes to the robot team.

Index Terms—Asynchronous communication, autonomous agents, distributed algorithms, partitioning algorithms.

I. INTRODUCTION

IN applications, such as environmental monitoring [1] or warehouse logistics [2], a team of robots is asked to perform tasks over a large space. The distributed *environment partitioning problem* consists of designing control and communication laws for individual robots such that the team divides a space into regions in order to optimize the quality of service provided. *Coverage control* additionally optimizes the positioning of robots inside a region. Coverage control and territory partitioning have applications in many fields. In cyber-physical systems, applications include automated environmental monitoring [1], fetching and delivery [2], and other vehicle-routing scenarios [3].

A broad discussion of partitioning and coverage control is presented in [4] which builds on the classic work of Lloyd [5] on algorithms for optimal quantizer design through “centering and partitioning.” The Lloyd-type approach was first adapted

for distributed coverage control in [6] and has since seen many variations, including nonconvex environments [7], [8] and self-triggered coverage algorithms [9].

Many existing coverage control algorithms assume that robots can communicate peer to peer [6], but in some environments, this is impractical. For example, underwater acoustic communication between ocean gliders has very low bandwidth and hilly or urban terrain can block radio communication. Instead, we present a coverage control algorithm for a team of robots which collectively maintains complete coverage of the environment and individually has only occasional contact with a central base station. This *one-to-base-station* communication model can represent ocean gliders surfacing to communicate with a tower [10], unmanned aerial vehicle (UAV) data mules that periodically visit ground robots [11], or cost-mindful use of satellite or cellular communication. Our algorithm optimizes the response time of the team to service requests in a nonconvex environment represented by a graph, with optimality defined by relevant “multicenter” cost functions for overlapping territories. Early work in coverage control of discrete nonconvex domains (represented by graphs) is presented in [12]. Discrete coverage problems are closely related to the literature on data clustering and k -means [13] as well as the facility location or k -center problem [14].

There are several specific contributions of this paper. First, we present the first coverage control algorithm for an asynchronous one-to-base-station communication model. This model is realistic and relevant for a variety of application domains. We handle the time delay between when robots communicate with the base station using overlapping regions instead of a partition. The algorithm can be adapted for various cost functions and allows for heterogeneity among agents. Second, we prove that the algorithm converges to a centroidal Voronoi partition in finite time for two relevant cost functions. Our Lyapunov argument is based on an adaptation of the standard partition-based coverage cost function. Third, we introduce the notion of Pareto-optimal partitions and provide a cost function to achieve such a partition using our algorithm. Finally, we describe how the algorithm can seamlessly handle changes in the environment as well as unscheduled arrival, departure or change in functionality of robots from the team. This feature leverages overlapping regions and eases integration of coverage control with task servicing.

A preliminary version of this paper appeared at the 2011 IEEE Conference on Decision Control and European Control Conference as [15]. This paper contains various results not found in [15]. First, we introduce a more general algorithm to

Manuscript received January 2, 2014; revised May 20, 2014; accepted April 9, 2015. Date of publication April 30, 2015; date of current version March 16, 2016. This work was supported in part by the National Science Foundation under Grant CPS-1035917 and in part by ARO under W911NF-11-1-0092. Recommended by Senior Editor Naomi Ehrlich Leonard and Associate Editor Fabio Fagnani.

R. Patel and F. Bullo are with the Department of Mechanical Engineering, University of California, Santa Barbara, CA 93106 USA (e-mail: r_patel@engineering.ucsb.edu; bullo@engineering.ucsb.edu).

P. Frasca is with the Department of Applied Mathematics, University of Twente, Enschede 7522 NB, The Netherlands (e-mail: p.frasca@utwente.nl).

J. W. Durham is with Kiva Systems, North Reading, MA 01864 USA (e-mail: joey@engineering.ucsb.edu).

R. Carli is with the Department of Information Engineering, University of Padova, Padova 35131, Italy (e-mail: carlirug@dei.unipd.it).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCNS.2015.2428304

achieve centroidal Voronoi partitions with a one-to-base-station communication architecture; we accomplish this by revising how updates occur in the algorithm. Second, we extend the framework of [15] by allowing for heterogeneous agents using weighted Voronoi partitions. Third, we introduce the notion of Pareto-optimal partitions and provide a method to converge to such a partition. Finally, simulation results have been extended and updated to demonstrate the performance of the new policies not present in [15].

II. PRELIMINARIES: COVERINGS OF GRAPHS AND COST FUNCTIONS

The one-to-base-station communication model studied in this paper requires that in the design of coverage algorithms, we adopt overlapping coverings instead of partitions. In this section, we translate concepts used in the partitioning of continuous environments [4] to coverings on graphs. In our notation, $\mathbb{R}_{>0}$, $\mathbb{R}_{\geq 0}$, and $\mathbb{Z}_{\geq 0}$, respectively, denote the sets of positive, non-negative and non-negative integer numbers. Given a set A , $|A|$ denotes the number of elements in A . Given sets A, B , their union and intersections are denoted as $A \cup B$ and $A \cap B$, respectively, and their difference is $A \setminus B = \{a \in A | a \notin B\}$.

A. Graphs and Distances

Let the finite set Q be a set of points in a continuous environment. These points can represent locations or small areas of interest. They are assumed to be the nodes of an (undirected) weighted graph $G(Q) = (Q, E, l)$ with edge set $E \subset Q \times Q$ and weight map $l : E \rightarrow \mathbb{R}_{>0}$; we let $l_e > 0$ be the weight of edge e . We assume that $G(Q)$ is connected and think of the edge weights as travel distances between nearby nodes.

In any weighted graph $G(Q)$, there is a standard notion of distance between vertices defined as follows. A *path* in G is an ordered sequence of vertices such that any consecutive pair of vertices is an edge of G . The *weight of a path* is the sum of the weights of the edges in the path. Given vertices h and k in G , the *distance* between h and k , denoted as $d_G(h, k)$, is the weight of the lowest weight path between them, or $+\infty$ if there is no path. If G is connected, then the distance between any two vertices is finite. By convention, $d_G(h, k) = 0$ if $h = k$. Note that $d_G(h, k) = d_G(k, h)$, for any $h, k \in Q$.

B. Coverings of Graphs

We will be covering Q with n subsets or regions which will each be owned by an individual agent.

Definition 1 (*n*-Covering): Given the graph $G(Q) = (Q, E, l)$, we define a *n*-covering of Q as a collection $P = \{P_i\}_{i=1}^n$ of subsets of Q such that:

- i) $\bigcup_{i=1}^n P_i = Q$;
- ii) $P_i \neq \emptyset$ for all $i \in \{1, \dots, n\}$.

Let $\text{Cov}_n(Q)$ be the set of *n*-coverings of Q .

Note that a vertex in Q may belong to multiple subsets in P , that is, a vertex may be covered by multiple agents. The above definition is an important change from prior work [12], which was limited to partitions of Q , defined as follows.

Definition 2 (*n*-Partition): A *n*-partition is a *n*-covering with the additional property that:

- iii) if $i \neq j$, then $P_i \cap P_j = \emptyset$.

Let $\text{Part}_n(Q)$ to be the set of *n*-partitions of Q .

Among the ways of covering Q , there is one which is worth special attention. Before we state the partition, let us define the vector of weights $w := \{w_1, \dots, w_n\}$, such that $w_i > 0$ and $\sum_{j=1}^n w_j = 1$. For brevity, we denote $\mathcal{W} = \{w \in \mathbb{R}_{>0}^n | \sum_{i=1}^n w_i = 1\}$. Then, given $w \in \mathcal{W}$ and a vector of distinct points $c \in Q^n$, the partition $P \in \text{Part}_n(Q)$ is said to be a *multiplicatively weighted Voronoi partition* of Q generated by c and weighted by w if, for each P_i and all $k \in P_i$, we have $c_i \in P_i$ and $1/w_i d_G(k, c_i) \leq (1/w_j) d_G(k, c_j)$, for $j \neq i$. The elements of c are said to be the generators of the Voronoi partition multiplicatively weighted by w . Note that there can be more than one multiplicatively weighted Voronoi partition generated by c and w since how to assign tied vertices is unspecified. The multiplicatively weighted Voronoi partition allow us to accommodate heterogeneous agents. For example, if agent i is faster than another agent j (i.e., $w_i > w_j$), it would make sense that agent i should control more territory than agent j . From this point forward, we refer to multiplicatively weighted Voronoi partitions simply as *Voronoi partitions* and the vector of weights w is given and fixed. Given that the weights are fixed, for the rest of this paper, we refer to a Voronoi partition generated by c and w simply as a Voronoi partition generated by c .

C. Cost Functions

Let *weight function* $\phi : Q \rightarrow \mathbb{R}_{>0}$ be a bounded positive function which assigns a relative weight to each element of Q . The weight assigned to a node by ϕ can be thought of as the “service time” or importance of that node. The *one-center function* \mathcal{H}_1 gives the cost for a robot to cover a subset $A \subset Q$ from a vertex $h \in A$ with relative prioritization given by ϕ

$$\mathcal{H}_1(h; A) = \sum_{k \in A} d_G(h, k) \phi(k).$$

This cost function leads us to the following definition.

Definition 3 (*Centroid*): We define the set of generalized centroids of $A \subset Q$ as the set of vertices in A which minimize \mathcal{H}_1 , that is

$$C(A) := \underset{h \in A}{\operatorname{argmin}} \mathcal{H}_1(h; A).$$

In what follows, we drop the word “generalized” for brevity. Note that the centroid of a set always belongs to the set. Fig. 1 shows an illustrative example of the set $C(A)$ for a simple environment.

With these notions, we are ready to define other useful cost functions. We can define the multicenter function $\mathcal{H} : Q^n \times \text{Cov}_n(Q) \rightarrow \mathbb{R}_{\geq 0}$ to measure the cost for n robots to cover a *n*-covering P from the vertices $c \in Q^n$ by

$$\mathcal{H}(c, P) = \sum_{i=1}^n \sum_{k \in P_i} \frac{1}{w_i} d_G(c_i, k) \phi(k).$$

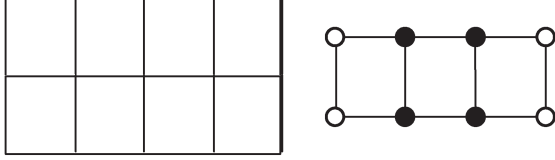


Fig. 1. Left image shows a grid environment whose corresponding graph representation is shown in the right image. Each cell in the grid represents a node in the graph and if two cells are adjacent, then there is a unit weight edge between those nodes. The black nodes in the graph denote the set of generalized centroids for the corresponding grid environment.

Note that if $w_i = w_j$ for all i, j , then the multicenter cost function above is the same as in [12]. Furthermore, we define the minimum cost-to-cover mapping $\mathcal{H}_{\min} : Q^n \times \text{Cov}_n(Q) \rightarrow \mathbb{R}_{\geq 0}$ by

$$\mathcal{H}_{\min}(c, P) = \sum_{k \in Q} \min_i \left\{ \frac{1}{w_i} d_G(c_i, k) \mid k \in P_i \right\} \phi(k).$$

Note that if P is a partition, then $\mathcal{H}_{\min}(c, P) = \mathcal{H}(c, P)$ for any c . We aim to minimize these performance functions with respect to the covering P and the vertices c . In the motivational scenario we are considering, each robot will be periodically asked to perform a task somewhere in its region with tasks located according to distribution ϕ . When idle, the robots would position themselves at the vertices c . By minimizing the coverage cost, the robot team minimizes the expected distance between a task and the furthest robot which can service the task.

We are almost ready to introduce a notion of optimal partition, the centroidal Voronoi partition. Our discussion begins with the following results, which are direct consequences of the above definitions.

Proposition 4 (Properties of \mathcal{H}): Let $P \in \text{Part}_n(Q)$ and $c \in Q^n$, then the following properties hold:

- i) If P' is a Voronoi partition generated by c , then

$$\mathcal{H}(c, P') \leq \mathcal{H}(c, P).$$

- ii) Let $\mathcal{I} \subset \{1, \dots, n\}$. If $c' \in Q^n$ satisfies $c'_i \in C(P_i)$ for $i \in \mathcal{I}$ and $c'_j = c_j$ for $j \notin \mathcal{I}$, then

$$\mathcal{H}(c', P) \leq \mathcal{H}(c, P)$$

with a strict inequality if $c_i \notin C(P_i)$ for any $i \in \mathcal{I}$.

Proposition 5 (Properties of \mathcal{H}_{\min}): Let $P' \in \text{Part}_n(Q)$ be a Voronoi partition generated by $c \in Q^n$, then the following properties hold:

- i) If $P \in \text{Cov}_n(Q)$ is a covering such that $P'_i \subseteq P_i$ for all i , then

$$\mathcal{H}_{\min}(c, P') = \mathcal{H}_{\min}(c, P).$$

- ii) If $\bar{P} \in \text{Part}_n(Q)$ is a partition satisfying $\bar{P}_i \cap P'_i \ni c_i$ for all i , then

$$\mathcal{H}_{\min}(c, P') \leq \mathcal{H}_{\min}(c, \bar{P}).$$

- iii) Let $\mathcal{I} \subset \{1, \dots, n\}$. If $c' \in Q^n$ satisfies $c'_i \in C(P'_i)$ for $i \in \mathcal{I}$ and $c'_j = c_j$ for $j \notin \mathcal{I}$, then

$$\mathcal{H}_{\min}(c', P') \leq \mathcal{H}_{\min}(c, P')$$

with a strict inequality if $c_i \notin C(P'_i)$ for any $i \in \mathcal{I}$.

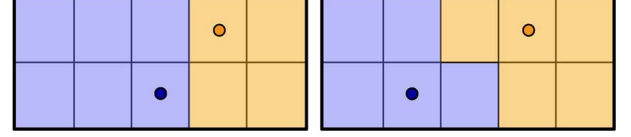


Fig. 2. Figure shows two environments with two agents. Each cell denotes a node in a graph and if two cells are adjacent, then there is a unit-weight edge between those nodes. The left image shows a Voronoi partition generated by the two agents. Note that the blue agent is not at its region's centroid. The right image is instead a centroidal Voronoi partition.

Propositions 4 and 5 imply that if $P \in \text{Part}_n(Q)$ and (c, P) minimizes \mathcal{H} (equivalently \mathcal{H}_{\min}), then $c_i \in C(P_i)$ for all i and P must be a Voronoi partition generated by c . This motivates the following definition.

Definition 6 (Centroidal Voronoi Partition): $P \in \text{Part}_n(Q)$ is a *centroidal Voronoi partition* of Q if there exists a $c \in Q^n$ such that P is a Voronoi partition generated by c and $c_i \in C(P_i)$ for all i .

For a given environment Q , a pair made of a centroidal Voronoi partition and the corresponding vector of centroids is locally optimal in the following sense: the cost functions \mathcal{H} and \mathcal{H}_{\min} cannot be reduced by changing either P or c independently. Fig. 2 demonstrates the difference between a Voronoi and centroidal Voronoi partition.

III. MODEL, PROBLEM, AND PROPOSED SOLUTION

A. One-to-Base-Station Robotic Network Model

Given a team of n robotic agents and a central base station, each agent $i \in \{1, \dots, n\}$ is required to have the following basic computation capabilities:

- (C1) agent i can identify itself to the base station;
- (C2) agent i has a processor with the ability to store a region $S_i \subset G(Q)$ and a center $s_i \in S_i$.

Each $i \in \{1, \dots, n\}$ is assumed to communicate with the base station according to the *asynchronous one-to-base-station communication model* described as follows:

- (C3) there exists a finite upper bound Δ on the time between communications between i and the base station. For simplicity, we assume no two agents communicate with the base station at the same time.

The base station must have the following capabilities:

- (C4) it can store an arbitrary n -covering of Q , $P = \{P_i\}_{i=1}^n$, a list of centroids $c \in Q^n$ and weights $w \in \mathcal{W}$;
- (C5) it can perform computations on subgraphs of $G(Q)$;
- (C6) it can store and operate on multiple n -coverings of Q , $P = \{P_i\}_{i=1}^n$, and a list of centroids $c \in Q^n$.

B. Problem Statement

Given weights $w \in \mathcal{W}$ assume that, for all $t \in \mathbb{R}_{\geq 0}$, each agent $i \in \{1, \dots, n\}$ maintains in memory a subset $S_i(t)$ of environment Q and a vertex $s_i(t) \in S_i(t)$. Our goal is to iteratively update the covering $S(t) = \{S_i\}_{i=1}^n$ and the centers $s(t) = \{s_i\}_{i=1}^n$ while solving the optimization problem

$$\min_{s \in Q^n} \min_{S \in \text{Cov}_n(Q)} U(s, S) \quad (1)$$

for some cost function $U(s, S)$ subject to the constraint that every node in the environment Q maintains coverage from some agent, and subject to the constraint imposed by the robot network model with asynchronous one-to-base-station communication.

C. One-to-Base Coverage Algorithm

Given the cost function defined by $U(s, S)$ and the one-to-base network model described by C1)–C6), we introduce the *One-to-Base Coverage Algorithm* to solve the optimization problem (1).

One-to-Base Coverage Algorithm

The base station maintains, in memory, an n -covering $P = \{P_i\}_{i=1}^n$, vector of locations $c = (c_i)_{i=1}^n$ and normalized weights $w = (w_i)_{i=1}^n$, while each robot i maintains, in memory, a set S_i and a vertex s_i . The base station maintains in temporary memory n -coverings $\bar{P} = \{\bar{P}_i\}_{i=1}^n$ and $\bar{\bar{P}} = \{\bar{\bar{P}}_i\}_{i=1}^n$, along with vectors $\bar{c} = (\bar{c}_i)_{i=1}^n$ and $\bar{\bar{c}} = (\bar{\bar{c}}_i)_{i=1}^n$ for computational purposes. At $t = 0$, let $P(0) \in \text{Cov}_n(Q)$, $S(0) = P(0)$, and let all $c_i(0)$'s be distinct. Assume that at time $t \in \mathbb{R}_{>0}$, robot i communicates with the base station. Let P^+ , c^+ , S_i^+ , and s_i^+ be the values after communication. Then, the base station executes the following actions:

- 1: update $\bar{P} := P$, $\bar{c} := c$, $\bar{\bar{P}} := P$, $\bar{\bar{c}} := c$
- 2: compute sets

$$P_{i,+} := \left\{ x \in Q \mid \frac{1}{w_i} d_G(x, c_i) < \min \left\{ \frac{1}{w_j} d_G(x, c_j) \mid x \in P_j, j \neq i \right\} \right\}$$

$$P_{i,-} := \left\{ x \in P_i \cap (\cup_{j \neq i} P_j) \mid \frac{1}{w_i} d_G(x, c_i) \geq \min \left\{ \frac{1}{w_j} d_G(x, c_j) \mid x \in P_j, j \neq i \right\} \right\}$$

- 3: update $\bar{\bar{P}}_i := (P_i \setminus P_{i,-}) \cup P_{i,+}$
- 4: **for** $k \in P_i \setminus c$ **do**
- 5: compute sets

$$P_{i,+} := \left\{ x \in Q \mid \frac{1}{w_i} d_G(x, k) < \min \left\{ \frac{1}{w_j} d_G(x, c_j) \mid x \in P_j, j \neq i \right\} \right\}$$

$$P_{i,-} := \left\{ x \in P_i \cap (\cup_{j \neq i} P_j) \mid \frac{1}{w_i} d_G(x, k) \geq \min \left\{ \frac{1}{w_j} d_G(x, c_j) \mid x \in P_j, j \neq i \right\} \right\}$$

- 6: update $\bar{\bar{P}}_i := (P_i \setminus P_{i,-}) \cup P_{i,+}$
 - 7: update $\bar{\bar{c}}_i := k$
 - 8: **if** $U(\bar{\bar{c}}, \bar{\bar{P}}) < U(\bar{c}, \bar{P})$ **then**
 - 9: update $\bar{\bar{P}}_i := \bar{\bar{P}}_i$
 - 10: update $\bar{\bar{c}}_i := \bar{\bar{c}}_i$
 - 11: $P_i^+ := \bar{\bar{P}}_i$
 - 12: $c_i^+ := \bar{\bar{c}}_i$
 - 13: tell agent i to set $S_i^+ := P_i^+$ and $s_i^+ = c_i^+$.
-

Remark 7 (Constant Cost): Given the constant cost function $U(c, P) = \alpha$ for $\alpha \in \mathbb{R}$, for a set of initial conditions (c, P) , the One-to-Base Coverage Algorithm produces a Voronoi partition generated by c .

Remark 8 (Full coverage): Notice that the set $P_{i,+}$ adds points to an agents environment from the other agent's territory that are closer to it. Also, notice that $P_{i,-}$ only removes points from agent i 's territory if another agent is covering that territory. Defining the sets in this way ensures that every point in the environment will always have coverage by some agent.

We have the following main result on the limit behavior of the algorithm.

Theorem 9 (Convergence of One-to-Base Coverage Algorithm (\mathcal{H}_{\min})): Consider a network consisting of n robots endowed with the computation capacities (C1), (C2) and communication capacity (C3), and a base station with capacities (C4), (C5), and (C6). Assume the network implements the One-to-Base Coverage Algorithm with $U(c, P) = \mathcal{H}_{\min}(c, P)$. Then, the resulting evolution

$$(s, S) : \mathbb{R}_{\geq 0} \rightarrow Q^n \times \text{Cov}_n(Q)$$

converges in finite time to a pair (s^*, S^*) composed of a centroidal Voronoi partition S^* generated by s^* .

D. Pareto-Optimal Partitions

Using the algorithms described thus far, ties along partitions' boundaries are not handled in any optimal way and can often be improved. The major source of suboptimal boundary allocation is due to the discrete nature of how centroids of a region are selected. Often times, when an agent has more than one "center" location, the overall partition can become better balanced if the agent takes an alternate center value as its centroid. The following definition and proposition make this notion more precise.

Definition 10 (Pareto-Optimal Partition): Given a vector of positions $c = \{c_1, \dots, c_n\}$, the Voronoi partition P generated by c is *Pareto-optimal* if for all $\bar{c} = \{c_1, \dots, \bar{c}_i, \dots, c_n\}$ for $i \in \{1, \dots, n\}$ such that $\bar{c}_i \neq c_i$ and $\bar{c}_i \in P_i$, the Voronoi partition \bar{P} generated by \bar{c} satisfies $\mathcal{H}(c, P) \leq \mathcal{H}(\bar{c}, \bar{P})$.

As an immediate consequence of the definition of a Pareto-optimal partition and of Proposition 4, we can conclude that *every Pareto-optimal partition is also a centroidal Voronoi partition*. However, the reverse implication does not hold, as shown in the following example.

Example 1 (1-D Pareto-optimal Partition): Consider the three environments in Fig. 3, each with two agents denoted by the colored circles. Assume that each cell denotes a node in the graph and that unit-weight edges connect any adjacent cells. Assume ϕ is constant. If the environment is partitioned according to leftmost image of Fig. 3, then each agent is at the centroid of its region, and the graph is a centroidal Voronoi partition whose multicenter function cost-to-cover is $\mathcal{H} = 4$. This partition is clearly not well balanced, and unless ties are broken in some nontrivial way, this is a valid (worst-case) partition that the system can reach. If, however, the blue agent moves to its other centroid, as shown in the middle image, then

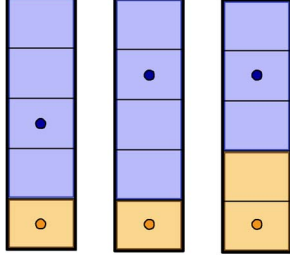


Fig. 3. Figure shows three environments with two agents. Each cell denotes a node in a graph, and if two cells are adjacent, then there is an unit-weight edge between those nodes.

the worst case partition must be a variant of the partition shown in the rightmost image whose cost-to-cover is $\mathcal{H} = 3$. There exists no partition with smaller cost-to-cover (with regards to \mathcal{H}) by moving any single agent and, hence, the partition in the rightmost image is Pareto-optimal.

The aforementioned results show that Pareto-optimal partitions are a subset of centroidal Voronoi partitions. We can define the cost function $U(c, P)$ in the One-to-Base Coverage Algorithm so that the algorithm converges to a Pareto-optimal partition. We define the new cost function

$$\mathcal{H}_{\text{inf}}(c) = \sum_{k \in Q} \min_i \left\{ \frac{1}{w_i} d_G(c_i, k) \mid k \in Q \right\} \phi(k). \quad (2)$$

Notice that this function is different from \mathcal{H}_{min} in that it looks for the absolute minimum distance to a point k . The function \mathcal{H}_{inf} allows the case when $k \notin P_i$, but $(1/w_i)d_G(c_i, k) < (1/w_j)d_G(c_j, k)$, for all $j \neq i$. The \mathcal{H}_{inf} function is linked to the multicenter function in the following sense.

Proposition 11 (Properties of \mathcal{H}_{inf}): Given $c \in Q^n$ and $w \in \mathcal{W}$, let P be a Voronoi partition generated by c , then

$$\mathcal{H}(c, P) = \mathcal{H}_{\text{inf}}(c).$$

Proof: Voronoi partitions are optimal in the sense that $\mathcal{H}(c, P) = \mathcal{H}_{\text{inf}}(c)$ by definition of \mathcal{H}_{inf} . ■

We are now ready to state the main result of this subsection. Given the *One-to-Base Coverage Algorithm* with $U(s, S) = \mathcal{H}_{\text{inf}}(c)$, we have the following result.

Theorem 12 (Convergence of One-to-Base Coverage Algorithm (\mathcal{H}_{inf})): Consider a network consisting of n robots endowed with the computation capacities (C1), (C2) and communication capacity (C3), and a base station with capacities (C4), (C5), and (C6). Assume that the network implements the One-to-Base Coverage Algorithm with $U(c, P) = \mathcal{H}_{\text{inf}}(c)$. Then, the resulting evolution

$$(s, S) : \mathbb{R}_{\geq 0} \rightarrow Q^n \times \text{Cov}_n(Q)$$

converges in finite time to a pair (s^*, S^*) composed of a Pareto-optimal partition S^* generated by s^* .

Some remarks are in order. First, it is possible for the One-to-base Algorithm with $U = \mathcal{H}_{\text{min}}$ to converge to a Pareto-optimal partition; however, it is not guaranteed as in the case with $U = \mathcal{H}_{\text{inf}}$. Second, if a partition is not Pareto-optimal, then the cost to cover a region, in the context of the multicenter function,

can be further decreased by making it Pareto-optimal. This point is clarified in Proposition 11, which relates the multicenter function to \mathcal{H}_{inf} . Finally, we emphasize that the difference in the cost-to-cover a region for a Pareto-optimal partition versus a centroidal Voronoi partition decreases as the map defining a region becomes less coarse. This is because the notion of a centroidal Voronoi partition not being Pareto-optimal only exists when a region has more than one centroid, a property of discrete spaces but not of continuous ones. Therefore, as the grid approximating a region becomes less coarse, the more likely it is that a centroidal Voronoi partition is also Pareto-optimal.

E. Combining \mathcal{H}_{min} and \mathcal{H}_{inf}

Although the One-to-Base Coverage Algorithm with $U = \mathcal{H}_{\text{inf}}$ is guaranteed to converge to a Pareto-optimal partition whereas the algorithm with $U = \mathcal{H}_{\text{min}}$ is not, the algorithm with $U = \mathcal{H}_{\text{min}}$ is still of practical importance. Computing \mathcal{H}_{inf} requires that for each node in the graph, all agents compare their relative distances to that node regardless of whether that node exists in the agent's territory or not. The \mathcal{H}_{min} function, however, only requires a comparison if the node belongs in the agent's territory. Given the computational capabilities of the base station being used, one method may be preferred over the other. A user can take advantage of both algorithm properties by running the algorithm with $U = \mathcal{H}_{\text{min}}$ until it converges to get an initial partition, and then run the algorithm with $U = \mathcal{H}_{\text{inf}}$ to reach a Pareto-optimal solution, if the solution system has not already reached it during the $U = \mathcal{H}_{\text{min}}$ portion of the algorithm. With a slight abuse of notation we will refer to this combined algorithm as the One-to-Base Coverage algorithm with $U = \mathcal{H}_{\text{min,inf}}$.

IV. IMPLEMENTATION AND SIMULATIONS

In order to efficiently implement the One-to-Base Coverage Algorithm and under the assumption of using \mathcal{H}_{min} or \mathcal{H}_{inf} as cost functions, we provide the following revised version, which can be easily seen to be equivalent to that in Section III-C.

One-to-Base Coverage Algorithm—revised

The base station maintains in memory an n -covering $P = \{P_i\}_{i=1}^n$, vector of locations $c = (c_i)_{i=1}^n$, and normalized weights $w = (w_i)_{i=1}^n$, while each robot i maintains, in memory, a set S_i and a vertex s_i . The base station maintains, in temporary memory, an n -covering $\bar{P} = \{\bar{P}_i\}_{i=1}^n$ and vectors $\bar{c} = (\bar{c}_i)_{i=1}^n$ and $\bar{w} = (\bar{w}_i)_{i=1}^n$ for computational purposes. At $t = 0$, let $P(0) \in \text{Cov}_n(Q)$, $S(0) = P(0)$, and let all $c_i(0)$'s be distinct. Assume that at time $t \in \mathbb{R}_{>0}$, robot i communicates with the base station. Let P^+, c^+, S_i^+ , and s_i^+ be the values after communication. Then, the base station executes the following actions:

- 1: update $\bar{P} := P$, $\bar{c} := c$, $\bar{w} := w$, $\bar{P}_i = Q$
- 2: **for** $k \in P_i \setminus c$ **do**
- 3: update $\bar{c}_i := k$
- 4: **if** $U(\bar{c}, \bar{P}) < U(\bar{c}, P)$ **then**
- 5: update $\bar{c}_i := k$

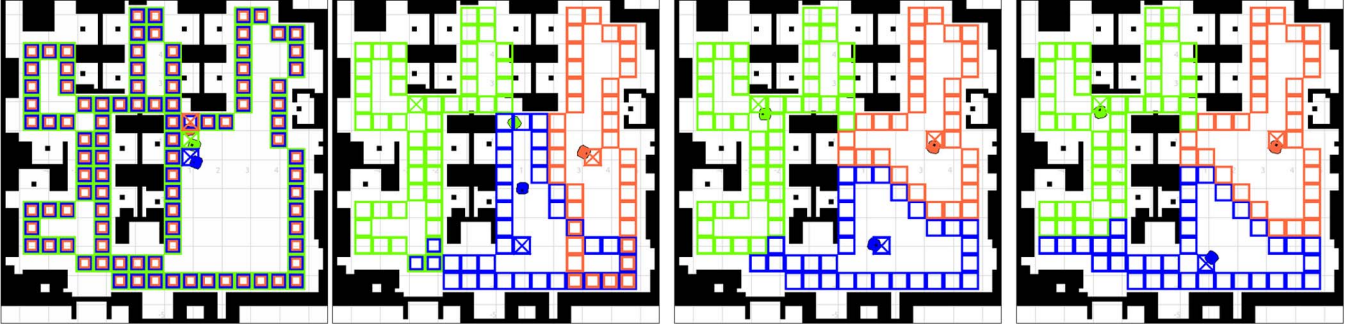


Fig. 4. Snapshots from a simulation of three robots partitioning an environment with black obstacles using the $\mathcal{H}_{\min, \inf}$ One-to-base station algorithm. The free space of the environment is modeled by using the indicated occupancy grid where each cell is a vertex in the resulting graph. The robots' optimal coverage position is marked by an X and the boundary of each robot's territory drawn in its color. Some cells are on the boundary of multiple territories: for these, we draw superimposed robot colors.

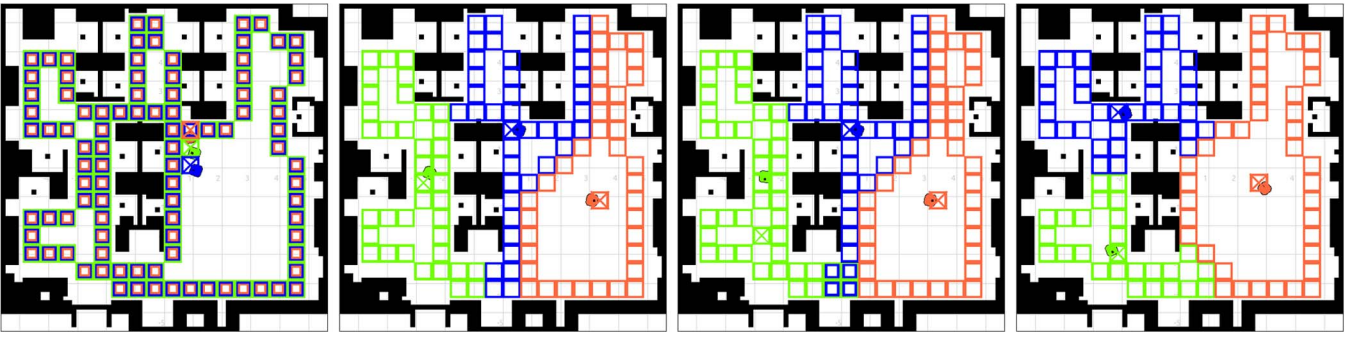


Fig. 5. Snapshots from a simulation of three robots partitioning an environment with black obstacles using the $\mathcal{H}_{\min, \inf}$ One-to-base station algorithm. Note that the initial condition is the same as in Fig. 4, but the evolution is different.

6: compute sets

$$\begin{aligned}
 P_{i,+} &:= \left\{ x \in Q \mid \frac{1}{w_i} d_G(x, \bar{c}_i) \right. \\
 &\quad \left. < \min \left\{ \frac{1}{w_j} d_G(x, c_j) \mid x \in P_j, j \neq i \right\} \right\} \\
 P_{i,-} &:= \left\{ x \in P_i \cap \left(\bigcup_{j \neq i} P_j \right) \mid \frac{1}{w_i} d_G(x, \bar{c}_i) \right. \\
 &\quad \left. \geq \min \left\{ \frac{1}{w_j} d_G(x, c_j) \mid x \in P_j, j \neq i \right\} \right\}
 \end{aligned}$$

7: $P_i^+ := (P_i \setminus P_{i,-}) \cup P_{i,+}$

8: $c_i^+ := \bar{c}_i$

9: tell agent i to set $S_i^+ := P_i^+$ and $s_i^+ = c_i^+$.

This revised version of the algorithm takes advantage of how the cost functions \mathcal{H}_{\min} and \mathcal{H}_{\inf} , and sets $P_{i,+}$ and $P_{i,-}$ are defined. Indeed, setting $\bar{P}_i = Q$ in line 1 avoids having to calculate $P_{i,+}$ and $P_{i,-}$ for every $k \in P_i \setminus c_i$, as was done in Section III-C, because \mathcal{H}_{\min} and \mathcal{H}_{\inf} already distribute costs to nodes that are closer to one agent as opposed to another. Hence, this implementation produces the same evolutions but requires less memory, as we no longer need the set \bar{P} , and less computation time, as the sets $P_{i,+}$ and $P_{i,-}$ are calculated only once.

We are now ready to proceed with our simulation results, which are obtained by running the revised version of the algorithm. To demonstrate the utility of the One-to-Base Coverage Algorithm for various values of cost function U , we implemented it using the open-source Player/Stage robot control

system and the Boost Graph Library (BGL). All results presented here are generated using Player 2.1.1, Stage 2.1.1, and BGL 1.34.1. A nonconvex environment (borrowed from [12]) is specified with three robots. The free space is modeled using an occupancy grid with 0.6 m resolution, producing a lattice-like graph with all edge weights equal to 0.6 m. The 0.6 m resolution is chosen so that each robot can fit in a grid cell.

One example with $U = \mathcal{H}_{\min, \inf}$ is shown in Fig. 4. In the simulation, the robots have a uniform weight assignment defined by $w_i = 1/3$ for $i \in \{1, \dots, n\}3$. We start with each robot owning the entire environment and stationed at its unique centroid as shown in the first panel, and then proceed by choosing a random robot to communicate with the base station at each iteration. The second panel shows an intermediate covering of the environment before convergence to a centroidal Voronoi partition. The third panel shows convergence of the $U = \mathcal{H}_{\min}$ portion of the $U = \mathcal{H}_{\min, \inf}$ algorithm. The fourth panel shows the Pareto-optimal partition which is achieved after convergence of the $U = \mathcal{H}_{\inf}$ portion of the $U = \mathcal{H}_{\min, \inf}$ algorithm. As can be seen, the movement of the robot relative to the third panel is marginal, but the partition appears to be more balanced and is still centroidal Voronoi. The cost to cover in terms of the multicenter function \mathcal{H} decreases from $\mathcal{H} = 729$ to $\mathcal{H} = 728$. Although the final partition and the decrease in cost-to-cover change only marginally in this example, the change can be much more noticeable as is explained in the following.

Another example with $U = \mathcal{H}_{\min, \inf}$ is shown in Fig. 5. As before, the robots have uniform weight assignment defined by $w_i = 1/3$ for $i \in \{1, \dots, n\}3$. The example starts with each

TABLE I
MULTICENTER FUNCTION COST-TO-COVER STATISTICS
FOR EACH ALGORITHM FROM 100 SIMULATION RUNS

Algorithm	Min	Mean	Max	StdDev
\mathcal{H}_{\min}	728	746.02	804	27.74
\mathcal{H}_{\inf}	728	730.26	732	1.92
$\mathcal{H}_{\min,\inf}$	728	730.38	753	4.91

agent owning the entire territory, with the agent being stationed at their unique centroid, and the simulation continuing with the agents being selected at random to communicate with the base station. The second panel shows the convergence of the $U = \mathcal{H}_{\min}$ portion of the algorithm, which leads to a final multicenter cost of $\mathcal{H} = 804$. The third panel shows the update after the first iteration of the $U = \mathcal{H}_{\inf}$ portion of the algorithm with the green agent. The update shows that the lower portion of the environment is getting less than optimal coverage and is improved by moving an agent closer to that region. The fourth panel shows the Pareto-optimal partition which is achieved after convergence of the $U = \mathcal{H}_{\inf}$ portion of the $U = \mathcal{H}_{\min,\inf}$ algorithm. Notice that in this example, the final partition is quite different from the partition achieved at the end of the $U = \mathcal{H}_{\min}$ portion of the algorithm. The final multicenter function cost of this partition is $\mathcal{H} = 753$, which is a noticeable improvement in coverage.

So far, we have looked at two representative examples of using the algorithm with $U = \mathcal{H}_{\min,\inf}$. These examples illustrate that, like centroidal Voronoi partitions, Pareto-optimal partitions are not necessarily unique, and that the evolution under the One-to-base station algorithm is only guaranteed to converge to a locally optimal solution. To see how the algorithm compares, in general, and for different choices of U , we simulate the algorithm with the same initial setup as shown in Figs. 4 and 5. The One-to-Base Coverage Algorithm with $U = \mathcal{H}_{\min}$, $U = \mathcal{H}_{\inf}$, and $U = \mathcal{H}_{\min,\inf}$ is run 100 times for each choice of U . Table I summarizes the final cost-of-coverage for each choice of U . We observe that the One-to-Base Coverage Algorithm with $U = \mathcal{H}_{\min}$ converges to partitions that have the same minimum cost as those attained with the algorithm using $U = \mathcal{H}_{\inf}$ or $U = \mathcal{H}_{\min,\inf}$. On the other hand, the maximum cost-to-cover with $U = \mathcal{H}_{\min}$ can be much larger than with the other two choices of U . Of the three algorithms, the algorithm with $U = \mathcal{H}_{\inf}$ converges consistently to partitions with the lowest coverage costs; however, as discussed earlier, it is computationally the most expensive. Finally, the algorithm with $U = \mathcal{H}_{\min,\inf}$ behaves as expected, converging on average to partitions with values similar to those of the algorithm with $U = \mathcal{H}_{\inf}$ although with a slightly larger deviation.

A. Handling Dynamic Changes

Evolving overlapping coverings in the One-to-Base Coverage Algorithm enable simple handling of environmental changes along with dynamic arrivals, departures, and even the disappearance of robots. Changes in the environment, along with robot departures or disappearances, can increase coverage cost, but those increases are only a transients and, with the appropriate algorithmic additions, the system will converge in finite steps after such an event. The One-to-Base Coverage Algorithm

also has the additional advantage that it can account for changes in robot performance due to changes in capability caused by potential damage to the hardware. The following algorithmic additions address how to handle the events described.

a) Environment changes: Each region in the environment is initially assigned an importance according to the weight function $\phi(x)$. As robots explore the environment, they may determine that certain regions are more/less important than what was originally assigned. Robots can communicate this to the base station at which point the base station can update $\phi(x)$.

b) Arrival: When a new robot i communicates with the base station, it can be assigned any initial P_i desired. Possibilities include adding all vertices within a set distance of its initial position or assigning it with just the single vertex which has the highest coverage cost in Q .

c) Departure and disappearance: A robot i might announce to the base station that it is departing, perhaps to recharge its batteries or to perform some other task. In this situation, the base station can simply add P_i to the territory of the next robot it talks to before executing the normal steps of the algorithm. The disappearance or failure of a robot i can be detected if it does not communicate with the base station for longer than Δ . If this occurs, then the departure procedure above can be triggered. Should i reappear later, it can be handled as a new arrival or given its old territory.

d) Performance: Malfunction of a robot i can be detected by the agent via self diagnosis and communicated to the base station. If this malfunction causes the robot to survey less territory, then w_i will have changed, so the base station can simply re-normalize the vector of weights w .

V. CONCLUSION

We have described a coverage algorithm, with corresponding cost functions, which uses the One-to-Base station communication architecture that drives territory ownership among a team of robots in a nonconvex environment to a centroidal Voronoi partition in finite time. We have also defined the notion of *Pareto-optimal partition* and have provided a provably correct method to reach such a partition using the One-to-Base Coverage Algorithm. Finally, we have demonstrated the effectiveness of the algorithm through simulation, and have outlined various ways that the algorithm can be adapted to allow for dynamic changes in the system. We have focused on dividing territory in this work, but the algorithm can easily be combined with methods to provide a service over Q , as in [16]. This work leaves various extensions open for further research. First, it would be worthwhile to adapt the algorithm to allow for area-constrained partitions similar to the work done in [17] and [18]. Second, we would like to extend the One-to-base Coverage Algorithm to other communication settings (e.g., directional or pair-wise gossip) to take advantage of the notion of Pareto-optimal partitions.

APPENDIX

In this appendix we prove Theorems 9 and 12. Any mention to the One-to-Base Coverage Algorithm in this appendix will

refer to the version presented in Section III-C. Their proof is based on the following convergence result for set-valued algorithms on finite state spaces, which can be recovered as a direct consequence of [19, Theor. 4.3].

Given a set X , a set-valued map $T : X \rightrightarrows X$ is a map which associates to an element $x \in X$ a subset $Z \subset X$. A set-valued map is nonempty if $T(x) \neq \emptyset$ for all $x \in X$. Given a nonempty set-valued map T , an evolution of the dynamical system associated to T is a sequence $\{x_n\}_{n \in \mathbb{Z}_{\geq 0}} \subset X$ with the property $x_{n+1} \in T(x_n)$ for all $n \in \mathbb{Z}_{\geq 0}$.

Lemma 3 (Convergence Under Persistent Switches): Let (X, d) be a finite metric space. Given a collection of maps $T_1, \dots, T_m : X \rightarrow X$, define the set-valued map $T : X \rightrightarrows X$ by $T(x) = \{T_1(x), \dots, T_m(x)\}$ and let $\{x_n\}_{n \in \mathbb{Z}_{\geq 0}}$ be an evolution of T . Assume that:

- (i) there exists a function $U : X \rightarrow \mathbb{R}$ such that $U(x') < U(x)$, for all $x \in X$ and $x' \in T(x) \setminus \{x\}$; and
- (ii) for all $i \in \{1, \dots, m\}$, there exists an increasing sequence of times $\{n_k | k \in \mathbb{Z}_{\geq 0}\}$ such that $x_{n_{k+1}} = T_i(x_{n_k})$ and $(n_{k+1} - n_k)$ is bounded.

Let $F_i = \{x \in X | T_i(x) = x\}$ be the set of fixed points of T_i . Then, for all $x_0 \in X$ there exist $N \in \mathbb{N}$ and $\bar{x} \in (F_1 \cap \dots \cap F_m)$ such that $x_n = \bar{x}$ for all $n \geq N$.

Note that the existence of a common fixed point for the collection of maps T_i is guaranteed by this result.

We now apply Lemma 13 to the evolution of One-to-base Coverage Algorithm with $U(c, P) = \mathcal{H}_{\min}(c, P)$ and $U(c, P) = \mathcal{H}_{\inf}(c)$, respectively. To do so, for each function given by $U(c, P)$, we must describe the algorithm as a set-valued map and find a corresponding Lyapunov function. The first step is possible because the One-to-Base Coverage Algorithm is well posed in the sense of the following immediate result.

Proposition 14 (Well Posedness): Let $P \in \text{Cov}_n(Q)$ and $c \in Q^n$ such that $c_i \in P_i$ and $c_i \neq c_j$ for all i and all $j \neq i$. Then, P^+ and c^+ produced by the *One-to-Base Coverage Algorithm* meet the same criteria.

Given this result, the One-to-Base Coverage Algorithm can be written as a set valued map. For any $i \in \{1, \dots, n\}$, we define the map $T_{U,i} : Q^n \times \text{Cov}_n(Q) \rightarrow Q^n \times \text{Cov}_n(Q)$ by

$$T_{U,i}(c, P) = \{\{c_1, \dots, c_i^+, \dots, c_N\}, \{P_1, \dots, P_i^+, \dots, P_n\}\}$$

where c_i^+ and P_i^+ are defined per the algorithm when i is the communicating robot, and U is dependent on the cost function we are referring to (i.e., $U = \mathcal{H}_{\min}$ or $U = \mathcal{H}_{\inf}$). Then, we can define the set-valued map $T_U : Q^n \times \text{Cov}_n(Q) \rightarrow Q^n \times \text{Cov}_n(Q)$ by

$$T_U(c, P) = \{T_{U,1}(c, P), \dots, T_{U,n}(c, P)\}.$$

Thus, the dynamical system defined by the application of the algorithm is described by $\{c^+, P^+\} \in T_U(c, P)$.

For our Lyapunov arguments, we will need to define $M(P)$ as the set of vertices which are owned by multiple agents. We now proceed by stating two useful propositions, which allow us to conclude Theorem 9.

Proposition 15 (Decaying \mathcal{H}_{\min} Cost Function): After each iteration of the one-to-base station algorithm if $(c^+, P^+) \neq (c, P)$ then one of the following holds:

- i) $\mathcal{H}_{\min}(c^+, P^+) < \mathcal{H}_{\min}(c, P)$;
- ii) $\mathcal{H}_{\min}(c^+, P^+) = \mathcal{H}_{\min}(c, P)$, and $|M(P^+)| < |M(P)|$.

Proof: If $c^+ = c$, then $\mathcal{H}_{\min}(c^+, P^+) \leq \mathcal{H}_{\min}(c, P)$. This is a direct consequence of how the sets $P_{i,+}$ and $P_{i,-}$ are defined. Points are added to P_i if and only if they are strictly closer to c_i than any other center c_j and, hence, the cost of \mathcal{H}_{\min} must decrease by the addition of these points. Points are removed if and only if they are strictly farther away or tied points and so \mathcal{H}_{\min} must decrease or stay the same. If $c^+ \neq c$, then by lines 8–10 of the algorithm $\mathcal{H}_{\min}(c^+, P^+) < \mathcal{H}_{\min}(c, P)$. For the case $\mathcal{H}_{\min} \text{in}(c^+, P^+) = \mathcal{H}_{\min} \text{in}(c, P)$, then for every $x \in P_j \setminus P_i$, for all $j \neq i$, there exists no point that is strictly closer to the center c_i than any other center $c_j, j \neq i$. Therefore, no points can be added to P_i^+ , and so if $P^+ \neq P$, it must be the case that $|M(P^+)| < |M(P)|$. ■

Proposition 16 (Convergence of $T_{\mathcal{H}_{\min}}$): The evolution of the One-to-Base Coverage Algorithm $(c(t), P(t))$ generated by the map $T_{\mathcal{H}_{\min}}$ converges in finite time to the intersection of the equilibria of the maps $T_{\mathcal{H}_{\min},i}$, that is, to a pair (c, P) where P is a centroidal Voronoi partition generated by c .

Proof: The proof proceeds with an application of Lemma 13 to $(c(t), P(t))$. The algorithm is the mapping $T_{\mathcal{H}_{\min}} : Q^n \times \text{Cov}_n(Q) \mapsto Q^n \times \text{Cov}_n(Q)$ defined above and is well posed. We can form a Lyapunov function using Proposition 15 as follows. Since the set Q is finite, there exists only a finite number of possible values for $\mathcal{H}_{\min} \text{in}$ and $|M|$. Let ϵ_m be the magnitude of the smallest nonzero difference between any two values of $\mathcal{H}_{\min} \text{in}$. Let α_M be larger than twice the maximum possible value of $|M|$. Define $V : Q^n \times \text{Cov}_n(Q) \rightarrow \mathbb{R}_{\geq 0}$ by

$$V(c, P) = \mathcal{H}_{\min}(c, P) + \frac{\epsilon_m}{\alpha_M} |M(P)|.$$

Thanks to this scaling of $|M(P)|$, Proposition 15 implies that if $(c', P') \in T_{\mathcal{H}_{\min}}(c, P)$, then either $V(c', P') < V(c, P)$ or $(c', P') = (c, P)$. Thus, $V(c, P)$ fulfills assumption (i) in Lemma 13. Moreover, the communication model (C3) ensures that assumption (ii) in Lemma 13 is met. Now, applying Lemma 13, we are ensured that the dynamics converge to a fixed point (c^*, P^*) . It remains to show that P^* is a centroidal Voronoi partition generated by c^* . We do this by refining in three sequential steps the properties that the fixed point must have: P^* is a partition; (c^*, P^*) is a Voronoi partition; and, finally, (c^*, P^*) is a centroidal Voronoi partition. First, if P^* is not a partition, then $P_{i,-}^* \neq \emptyset$; this establishes that P^* is a partition. Second, if the partition P^* is not Voronoi, then $P_{i,+}^* \neq \emptyset$; this establishes that (c^*, P^*) is a Voronoi partition. Third, if P^* is a Voronoi partition generated by c^* , but $c_i^* \notin C(P_i^*)$ for any i , then from part (iii) of Proposition 5, there exists a location c_i^{**} (at a centroid location) that improves the cost to cover P_i^* ; line 4 of the algorithm guarantees that this location is checked, and lines 8–10 ensure the position is updated to that location or one with an even lower cost. It should be noted that an update in location to c_i^{**} can simultaneously lead to an update in territory P_i^* : however, given that P^* is a partition, only $P_{i,+}^*$ can contribute to

the territory, which further decreases the coverage cost by the definition of \mathcal{H}_{\min} and $P_{i,+}^*$. Therefore, we have established that there exists an update reducing the cost function $T_{\mathcal{H}_{\min}}$ if the fixed point is not a centroidal Voronoi partition. Then, P^* must be a centroidal Voronoi partition generated by c^* . (Note that the fixed point could also potentially have other properties in addition to being a centroidal Voronoi partition, however, establishing those properties are beyond the scope of this proof.) ■

Since updates to agent i in base-station memory also occur on the physical agent, we can conclude the convergence proof of Theorem 9.

Finally, we state two propositions which allow us to conclude Theorem 12.

Proposition 17 (Decaying \mathcal{H}_{\inf} Cost Function): After each iteration of the one-to-base station algorithm if $(c^+, P^+) \neq (c, P)$ using \mathcal{H}_{\inf} as the cost function then one of the following holds:

- i) $\mathcal{H}_{\inf}(c^+) < \mathcal{H}_{\inf}(c)$;
- ii) $\mathcal{H}_{\inf}(c^+) = \mathcal{H}_{\inf}(c)$, and $\mathcal{H}_{\min}(c^+, P^+) < \mathcal{H}_{\min}(c, P)$;
- iii) $\mathcal{H}_{\inf}(c^+) = \mathcal{H}_{\inf}(c)$, $\mathcal{H}_{\min}(c^+, P^+) = \mathcal{H}_{\min}(c, P)$ and $|M(P^+)| < |M(P)|$.

Proof: If $c^+ = c$ then $\mathcal{H}_{\min}(c^+, P^+) \leq \mathcal{H}_{\min}(c, P)$ and $\mathcal{H}_{\inf}(c^+) = \mathcal{H}_{\inf}(c)$. This is a direct consequence of how we define \mathcal{H}_{\inf} and the sets $P_{i,+}$ and $P_{i,-}$. Points are added to P_i if and only if they are strictly closer to c_i than any other center c_j and hence the cost of \mathcal{H}_{\min} must decrease by the addition of these points. Points are removed if and only if they are strictly farther away or tied points and so \mathcal{H}_{\min} must decrease or stay the same. If $c^+ \neq c$ then by the lines 8–10 of the algorithm $\mathcal{H}_{\inf}(c^+) < \mathcal{H}_{\inf}(c)$. For the case $\mathcal{H}_{\min}(c^+, P^+) = \mathcal{H}_{\min}(c, P)$, then for every $x \in P_j \setminus P_i$, for all $j \neq i$, there exist no point that is strictly closer to the center c_i than any other center c_j , $j \neq i$. Therefore, no points can be added to P_i^+ , and so if $P^+ \neq P$ it must be the case that $|M(P^+)| < |M(P)|$. ■

Proposition 18 (Convergence of $T_{\mathcal{H}_{\inf}}$): The evolution of the One-to-Base Coverage Algorithm $(c(t), P(t))$ generated by the map $T_{\mathcal{H}_{\inf}}$ converges in finite time to the intersection of the equilibria of the maps $T_{\mathcal{H}_{\inf}, i}$, that is, to a pair (c, P) where P is a Pareto-optimal partition generated by c .

Proof: The proof follows the lines of the proof of Proposition 16, with the important modification of using a different Lyapunov function, defined as follows. Let ϵ_i and ϵ_m be the magnitude of the smallest possible nonzero difference between two values of \mathcal{H}_{\inf} and \mathcal{H}_{\min} , respectively. Let α_m and α_M be larger than twice the maximum possible value of \mathcal{H}_{\min} and $|M|$, respectively. If we define the function V as

$$V(c, P) = \mathcal{H}_{\inf}(c) + \frac{\epsilon_i}{\alpha_m} \mathcal{H}_{\min}(c, P) + \frac{\epsilon_i \epsilon_m}{\alpha_m \alpha_M} |M(P)|$$

and we invoke Proposition 17 and Lemma 13, we conclude that the dynamics converge to a fixed point (c^*, P^*) .

It remains to be shown that P^* is a Pareto-optimal partition generated by c^* . If P^* is not a partition, then $P_{i,-}^* \neq \emptyset$ and if the partition is not Voronoi, then $P_{i,+}^* \neq \emptyset$. Continuing by contradiction, assume that c^* forms a Voronoi partition which is not Pareto-optimal. This implies that there exists a $c' \in P^*$ where for at least one agent $c'_i \neq c_i^*$ and a partition P' generated

by c' such that $\mathcal{H}(c', P') < \mathcal{H}(c^*, P^*)$. By the definition of the algorithm and Proposition 11, this is not possible. Therefore, the fixed point partition is Pareto-optimal. ■

REFERENCES

- [1] E. Fiorelli, N. E. Leonard, P. Bhatta, D. A. Paley, R. Bachmayer, and D. M. Fratantoni, "Multi-AUV control and adaptive sampling in Monterey Bay," *IEEE J. Ocean. Eng.*, vol. 31, no. 4, pp. 935–948, Oct. 2006.
- [2] P. R. Wurman, R. D'Andrea, and M. Mountz, "Coordinating hundreds of cooperative, autonomous vehicles in warehouses," *AI Mag.*, vol. 29, no. 1, pp. 9–20, 2008.
- [3] B. Golden, S. Raghavan, and E. Wasil, *The Vehicle Routing Problem: Latest Advances and New Challenges*, vol. 43, *Operations Research/Computer Science Interfaces*. New York, USA: Springer-Verlag, 2008.
- [4] F. Bullo, J. Cortés, and S. Martínez, *Distributed Control of Robotic Networks*. Princeton, NJ, USA: Princeton University Press, 2009.
- [5] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. IT-28, no. 2, pp. 129–137, Mar. 1982.
- [6] J. Cortés, S. Martínez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Trans. Robot. Autom.*, vol. 20, no. 2, pp. 243–255, Apr. 2004.
- [7] L. C. A. Pimenta, V. Kumar, R. C. Mesquita, and G. A. S. Pereira, "Sensing and coverage for a network of heterogeneous robots," in *Proc. IEEE Conf. Dec. Control*, Cancún, México, Dec. 2008, pp. 3947–3952.
- [8] S. Bhattacharya, R. Ghrist, and V. Kumar, "Multi-robot coverage and exploration in non-Euclidean metric spaces," in *Algorithmic Foundations of Robotics X*, vol. 86. New York, USA: Springer-Verlag, 2013, pp. 245–262.
- [9] C. Nowzari and J. Cortés, "Self-triggered coordination of robotic networks for optimal deployment," *Automatica*, vol. 48, no. 6, pp. 1077–1087, 2012.
- [10] A. Pereira, H. Heidarrson, C. Oberg, D. Caron, B. Jones, and G. Sukhatme, "A communication framework for cost-effective operation of AUVs in coastal regions," in *Field and Service Robotics*, vol. 62, *Tracts in Advanced Robotics*, A. Howard, K. Iagnemma, and A. Kelly, Eds. New York, USA: Springer-Verlag, 2010, pp. 433–442.
- [11] R. C. Shah, S. Roy, S. Jain, and W. Brunette, "Data MULEs: modeling and analysis of a three-tier architecture for sparse sensor networks," *Ad Hoc Netw.*, vol. 1, no. 2/3, pp. 215–233, 2003.
- [12] J. W. Durham, R. Carli, P. Frasca, and F. Bullo, "Discrete partitioning and coverage control for gossiping robots," *IEEE Trans. Robot.*, vol. 28, no. 2, pp. 364–378, Apr. 2012.
- [13] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, 1999.
- [14] V. V. Vazirani, *Approximation Algorithms*. New York, USA: Springer-Verlag, 2001.
- [15] J. W. Durham, R. Carli, P. Frasca, and F. Bullo, "Dynamic partitioning and coverage control with asynchronous one-to-base-station communication," in *Proc. IEEE Conf. Dec. Control Eur. Control Conf.*, Orlando, FL, USA, Dec. 2011, pp. 5589–5594.
- [16] F. Bullo, E. Frazzoli, M. Pavone, K. Savla, and S. L. Smith, "Dynamic vehicle routing for robotic systems," in *Proc. IEEE*, 2011, vol. 99, no. 9, pp. 1482–1504.
- [17] J. Cortés, "Coverage optimization and spatial load balancing by robotic sensor networks," *IEEE Trans. Autom. Control*, vol. 55, no. 3, pp. 749–754, Mar. 2010.
- [18] R. Patel, P. Frasca, and F. Bullo, "Centroidal area-constrained partitioning for robotic networks," *ASME J. Dynam. Syst. Meas., Control*, vol. 136, no. 3, p. 031024, 2014.
- [19] F. Bullo, R. Carli, and P. Frasca, "Gossip coverage control for robotic networks: Dynamical systems on the space of partitions," *SIAM J. Control Optimiz.*, vol. 50, no. 1, pp. 419–447, 2012.



Rushabh Patel (S'12) received the B.S. and M.S. degrees in aerospace engineering from California Polytechnic State University, San Luis Obispo, CA, USA, in 2007, and the Ph.D. degree in control and dynamical systems at the University of California, Santa Barbara, CA, USA, in 2015.

Currently, he is a Control Analyst at Northrop Grumman Aerospace Systems, Redondo Beach, CA, USA. From 2007 to 2011, he was a Performance Analyst with Northrop Grumman Aerospace Systems. His research interests include coordination and surveillance in networked systems.



Paolo Frasca received the Ph.D. degree in mathematics for engineering sciences from Politecnico di Torino, Torino, Italy, in 2009.

Between 2008 and 2013, he held research and visiting positions at the University of California, Santa Barbara, CA, USA, at the IAC-CNR, Rome, and at the Politecnico di Torino. Since 2013, he is an Assistant Professor with the Department of Applied Mathematics, University of Twente, Enschede, the Netherlands. He has (co)authored more than 40 journal and conference papers and has given several

invited talks at international institutes and events on his research interests: theory of network systems and cyber-physical systems, with applications on robotic, sensor, infrastructural, and social networks.

Prof. Frasca is a recipient of the 2013 SIAG/CST Best SICON Paper Prize. He is currently Subject Editor for the international journal of robust and nonlinear control and in the conference editorial boards of the IEEE Control Systems Society and of the European Control Association (EUCA).



Joseph W. Durham received the B.A. (Hons.) degree in physics from Carleton College, Northfield, MN, USA, in 2004, and the M.Sc. and Ph.D. degrees in control and dynamical systems from the University of California, Santa Barbara, CA, USA, in 2007 and 2011, respectively.

Currently, he is a Senior Research Scientist at Kiva Systems, North Reading, MA, USA. His research interests include robotic coordination and navigation, distributed resource allocation, and voting methods.



Ruggero Carli received the Laurea degree in computer engineering and the Ph.D. degree in information engineering from the University of Padova, Padova, Italy, in 2004 and 2008, respectively.

He was a Postdoctoral Fellow with the Department of Mechanical Engineering, University of California at Santa Barbara, Santa Barbara, CA, USA, from 2008 to 2010. Currently, he is an Assistant Professor with the Department of Information Engineering, University of Padova. He has published more than 70 journal and conference papers. His current research

interests include control theory and, in particular, control under communication constraints, cooperative control, and distributed estimation.



Francesco Bullo (S'95–M'99–SM'03–F'10) is a Professor with the Mechanical Engineering Department and the Center for Control, Dynamical Systems and Computation, at the University of California, Santa Barbara, CA, USA. He was previously with the University of Padova, Padova, Italy; the California Institute of Technology, Pasadena, CA, USA, and the University of Illinois at Urbana-Champaign, Champaign, IL, USA. He is the coauthor of *Geometric Control of Mechanical Systems* (Springer, 2004) and *Distributed Control of Robotic Networks*

(Princeton, 2009). His main research interests are network systems and distributed control with applications to robotic coordination, power grids, and social networks.

Prof. Bullo received the 2008 IEEE CSM Outstanding Paper Award, the 2010 Hugo Schuck Best Paper Award, the 2013 SIAG/CST Best Paper Prize, and the 2014 IFAC Automatica Best Paper Award. He has served on the editorial boards of IEEE TRANSACTIONS ON AUTOMATIC CONTROL, *ESAIM: Control, Optimization, and the Calculus of Variations*, *SIAM Journal of Control and Optimization*, and *Mathematics of Control, Signals, and Systems*.