10-25-2010

Adaptive Algorithms for Coverage Control and Space Partitioning in Mobile Robotic Networks

Jerome Le Ny
University of Pennsylvania, jerome.le-ny@polymtl.ca

George J. Pappas
University of Pennsylvania, pappasg@seas.upenn.edu

Adaptive Algorithms for Coverage Control and Space Partitioning in Mobile Robotic Networks

Abstract
We consider deployment problems where a mobile robotic network must optimize its configuration in a distributed way in order to minimize a steady-state cost function that depends on the spatial distribution of certain probabilistic events of interest. Three classes of problems are discussed in detail: coverage control problems, spatial partitioning problems, and dynamic vehicle routing problems. Moreover, we assume that the event distribution is a priori unknown, and can only be progressively inferred from the observation of the location of the actual event occurrences. For each problem we present distributed stochastic gradient algorithms that optimize the performance objective. The stochastic gradient view simplifies and generalizes previously proposed solutions, and is applicable to new complex scenarios, for example adaptive coverage involving heterogeneous agents. Finally, our algorithms often take the form of simple distributed rules that could be implemented on resource-limited platforms.

Keywords
robotics, coverage control problems, partitioning algorithms, stochastic gradient algorithms, dynamic vehicle routing problems, adaptive algorithms

Disciplines
Artificial Intelligence and Robotics | Controls and Control Theory | Operational Research | Robotics | Theory and Algorithms

Comments

Adaptive Algorithms for Coverage Control and Space Partitioning in Mobile Robotic Networks

Jerome Le Ny, *Member, IEEE,* and George J. Pappas, *Fellow, IEEE*

Abstract

We consider deployment problems where a mobile robotic network must optimize its configuration in a distributed way in order to minimize a steady-state cost function that depends on the spatial distribution of certain probabilistic events of interest. Three classes of problems are discussed in detail: coverage control problems, spatial partitioning problems, and dynamic vehicle routing problems. Moreover, we assume that the event distribution is a priori unknown, and can only be progressively inferred from the observation of the location of the actual event occurrences. For each problem we present distributed stochastic gradient algorithms that optimize the performance objective. The stochastic gradient view simplifies and generalizes previously proposed solutions, and is applicable to new complex scenarios, for example adaptive coverage involving heterogeneous agents. Finally, our algorithms often take the form of simple distributed rules that could be implemented on resource-limited platforms.

Index Terms

Coverage control problems, dynamic vehicle routing problems, partitioning algorithms, stochastic gradient descent algorithms, adaptive algorithms, potential field based motion planning.

I. Introduction

The deployment of large-scale mobile robotic networks has been an actively investigated topic in recent years [1]–[3]. Applications range from Intelligence, Surveillance and Reconnaissance

This work was supported by the ONR-MURI HUNT award N00014-08-1-0696.

The authors are with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104, USA. jeromel, pappasg@seas.upenn.edu.

missions for Unmanned Aerial Vehicles to environmental monitoring, search and rescue missions, and transportation and distribution tasks. With the increase in size of these networks, relying on human operators to remotely pilot each vehicle is becoming impractical. Attention is increasingly focusing on enabling autonomous operations, so that these systems can decide online how to concentrate their activities where they are most critical.

A mobile robotic network should have the capability of autonomously deploying itself in a region of interest to reach a configuration optimizing a given performance objective [3, chapter 5]. Such problems can be distinguished based on the deployment objective, and among them the coverage control problem introduced by Cortés et al. [4] has proved to be particularly important. In this problem, the quality of a given robot configuration is measured by a multicenter function from the locational optimization and vector quantization literature [5], [6]. A distributed version of the Lloyd quantization algorithm [7] allows a robotic network to locally optimize the utility function in a way that scales gracefully with the size of the network [4]. The asymptotic configuration forms a centroidal Voronoi partition [8] of the workspace. The basic version of the coverage control problem has inspired many variations, e.g. considering limited communication and sensing radii [9], [10], heterogeneous sensors [11], obstacles and non-point robots [12], or applications to field estimation problems [13]. It is also tightly connected to certain vehicle routing problems, notably the Dynamic Traveling Repairman Problem (DTRP) [14]–[16], as discussed by Frazzoli and Bullo in [17] and several subsequent papers, see e.g. [18], [19]. Another related problem is the space partitioning problem, see e.g. [20], [21], where the robots must autonomously divide the environment in order to balance the workload among themselves.

In essentially all the previously mentioned applications, the goal of the robotic network is to respond to events appearing in the environment. For example in the DTRP, jobs appear over time at random spatial locations and are serviced by the mobile robots traveling to these locations. The utility function optimized by the network invariably depends on the spatial probability distribution of the events, and the optimization algorithms require the knowledge of this distribution [4], [17], [20], [21]. Hence they are not applicable in the commonly encountered situations where the robots do not initially have such knowledge but can only observe the event locations over time. It is then natural to ask how to gradually improve the spatial configuration of the robotic network based only on the observation of the successive event locations. Recently, coverage control algorithms [22] and vehicle routing algorithms [19], [23] have been developed that work in the absence of a priori knowledge of the event distribution. We call these algorithms *adaptive*, in analogy with the engineering literature on adaptive systems [24]. A somewhat different problem is considered in [13], where the robots can directly measure the values of a field at their at their current positions, and then optimize a coverage objective for an estimate of this field.

Robotic deployment algorithms rely heavily on concepts and algorithms from geometric and locational optimization and vector quantization [3], see e.g [6], [25], [26] for general references on these topics. Indeed, Lloyd's algorithm [7] optimizes the least-squares coverage utility function [4]. Its adaptive version, also known as the K-means algorithm of MacQueen [27], the LBG algorithm [5], or Kohonen's 0-neighbor self-organizing map [28], is particularly related to the adaptive coverage control problem discussed in Section III. For example, our algorithm for the DTRP in light traffic can be viewed as a version of MacQueen's algorithm for an L^1 distortion measure. Non-adaptive partitioning algorithms have also been studied in the geometric optimization literature [29], [30]. In particular, Aurenhammer et al. [29] present a gradient descent based least-squares partitioning algorithm, which can be implemented in a distributed way in a robotic network [21].

Statement of Contributions: An essential idea of our work is that deployment problems with stochastic uncertainty can often be discussed from the unifying point of view of stochastic gradient algorithms, thereby clarifying the convergence proofs and allowing to easily derive new algorithms for complex problems. In this paper we restrict our attention to three related classes of problems: coverage control, spatial partitioning, and dynamic vehicle routing problems. For these three applications, we derive distributed stochastic gradient algorithms that optimize the utility functions *in the absence of a priori knowledge of the event distribution*. Remarkably, the algorithms we describe often take the form of simple rules, in fact typically simpler than the corresponding non-adaptive algorithms. Hence they are easier to implement on small platforms with constrained computational and communication capabilities.

Specifically, we first discuss in Section III certain stochastic gradient algorithms that adaptively optimize coverage control objectives. We can then easily derive algorithms for new complex multi-agent deployment problems and justify this claim by developing solutions to coverage control problems involving Markovian event dynamics or heterogeneous robots. Additional application examples, including deployment under realistic stochastic wireless connectivity constraints, can be found in [31]. In Section IV, we describe new adaptive distributed algorithms that partition the workspace between the robots in order to balance their workload, using only the observation of the past event locations. These algorithms exploit the link between generalized Voronoi diagrams and certain Monge-Kantorovich optimal transportation problems [32]–[34]. Finally in Section V we present an adaptive algorithm for the DTRP. In light traffic conditions, the algorithm reduces to the coverage control algorithm of Section III, and is simpler than the algorithm presented in [23]. In heavy traffic conditions, it relies on the partitioning algorithm of Section IV. This fully adaptive algorithm for the DTRP completes the recent work of Pavone et al. [19], whose algorithm requires the knowledge of the event distribution in the heavy traffic regime.

II. Preliminaries

A. Notation

We denote $[n] := \{1, \ldots, n\}$. Throughout the paper all random elements are defined on a generic probability space $(\Omega, \mathcal{F}, P)$. We abbreviate "independent and identically distributed" by iid. For $q \geq 1$, the Lebesgue measure of a set $A \subset \mathbb{R}^q$ is denoted $|A|$. A Borel measure μ on $\mathbb{R}^q$ is said to dominate the Lebesgue measure if $|A| = 0$ for all Borel sets A such that $\mu(A) = 0$. We denote the *Euclidean norm* on $\mathbb{R}^q$ by $\|\cdot\|$. Let (X, d) be a metric space. For a set $S \subset X$, we denote the indicator function of S by $\mathbf{1}_S$, i.e., $\mathbf{1}_S(x) = 1$ if $x \in S$ and $\mathbf{1}_S(x) = 0$ otherwise. For $x_0 \in X$, the Dirac measure at x_0 is denoted by δ_{x_0} and defined by $\delta_{x_0}(S) = 1_S(x_0)$ for all Borel subsets S of X. We denote the distance from a point $x \in X$ to a set S by $d_S(x) := d(x, S) := \inf_{y \in S} d(x, y)$, and we set $d(x, \emptyset) = +\infty$. A sequence of points $\{x_k\}_{k \geq 0}$ in a X is said to converge to a set $S \subset X$ if $d(x_k, S) \to 0$ as $k \to \infty$. For nonempty sets $B, C \subset X$, the Hausdorff pseudometric is defined by $d_H(B, C) := \max(\sup_{x \in B} d(x, C), \sup_{x \in C} d(x, B))$. The ball of radius r around $S \subset X$ is $B(S, r) := \{x \in X | d(x, S) \leq r\}$. Also $B(\{x\}, r)$ is just denoted $B(x, r)$.

B. Robot Network Model

We consider a group of n robots evolving in a workspace $\mathsf{Q} \subset \mathbb{R}^q$, for some $q \geq 1$. The set Q is assumed to be *compact convex with a non-empty interior*. We denote the robot positions at time $t \in \mathbb{R}_{\geq 0}$ by $p(t) = [p_1(t), \ldots, p_n(t)] \in \mathsf{Q}^n$. For simplicity, we assume that the robots follow a simple kinematic model

$$\forall i \in [n], \forall t \in \mathbb{R}_{\geq 0},\ \dot{p}_i(t) = u_i, \quad |u_i(t)| \leq v_i, \quad \text{with } v_i > 0, \tag{1}$$

where u_i is a bounded control input. However, more complex dynamics could be considered since our analysis only involves the positions of the robots at certain discrete times, see e.g. (13). In addition, the robots are assumed to perform computations and to communicate instantaneously.

We also define

$$\mathsf{D}_n = \left\{ x = [x_1^T, \ldots, x_n^T]^T \in (\mathbb{R}^q)^n \,\middle|\, x_i = x_j \text{ for some } i \neq j, 1 \leq i, j \leq n \right\}. \tag{2}$$

Hence $\mathsf{D}_n \cap \mathsf{Q}^n$ is the (unphysical) set of configurations where at least two robots occupy the same position.

C. Geometric Optimization

For a vector $p = [p_1, \ldots, p_n] \in (\mathbb{R}^q)^n \setminus \mathsf{D}_n$, we define the Voronoi cell of point p_i by

$$V_i(p) = \left\{ z \in \mathbb{R}^q \middle| \|z - p_i\| \leq \|z - p_j\|, \forall j \in [n] \right\}.$$

That is, V_i is the set of points in the workspace for which robot i is the closest robot *for the Euclidean distance*. The Voronoi cells of the points divide $\mathbb{R}^q$ into closed convex polyhedra, and $\{V_i\}_{i \in [n]}$ is called a Voronoi diagram [25]. Two points p_i and p_j or their indices i, j are called Voronoi neighbors if the boundaries of their Voronoi cells intersect, i.e., if $V_i(p) \cap V_j(p) \neq \emptyset$.

Now let $f : \mathbb{R}_{\geq 0} \to \mathbb{R}$ be an increasing function, $w = [w_1, \ldots, w_n] \in \mathbb{R}^n$, and $p = [p_1, \ldots, p_n] \in (\mathbb{R}^q)^n \setminus \mathsf{D}_n$. We define the generalized Voronoi cell of the pair (p_i, w_i) with respect to f by

$$V_i^f(p, w) = \left\{ z \in \mathbb{R}^q \middle| f(\|z - p_i\|) - w_i \leq f(\|z - p_j\|) - w_j, \forall j \in [n] \right\}. \tag{3}$$

The point p_i is called the generator and w_i the weight of the cell $V_i^f(p, w)$, and $\{V_i^f\}_{i \in [n]}$ a generalized Voronoi diagram. In particular for $f(x) = x^2$, the generalized Voronoi diagram is called a power diagram [25], [35], and the generalized Voronoi cell a power cell. Power cells are also (possibly empty) polyhedra, but this property is not true in general for generalized Voronoi diagrams. Clearly, a generalized Voronoi diagram is a Voronoi diagram if and only if all pairs have the same weight $w_i = w_j, \forall i, j \in [n]$. In general, the size of a generalized Voronoi cell of a pair increases as its weight increases with respect to the weights of the other pairs. Similarly to Voronoi neighbors, we define generalized Voronoi neighbors and power diagram neighbors.

D. Min-consensus

At several occasions, we need to solve the following problem in a distributed manner in the robotic network. Robot i, for $i \in [n]$, is associated to a certain quantity $\hat{d}_i \in \mathbb{R}$, which can be $+\infty$. Each robot must decide if it belongs to $\arg\min_{i\in[n]} \hat{d}_i$. For simplicity, we assume that each robot can communicate with some other robots along bidirectional links in such a way that the global communication network is connected. We also assume that the robots know the diameter of the network, denoted $\mathtt{diam}$. Alternatively, they know the number n of robots in the system, in which case we take $\mathtt{diam} = n$ below.

In a synchronous network the problem can be solved by the $\mathtt{FloodMin}$ algorithm [36, section 4.1.2]. Every robot maintains a record in a variable $\mathtt{d_i}$ of the minimum number it has seen so far, with $\mathtt{d_i} = \hat{d}_i$ initially. At each round, the process sends this minimum to all its neighbors. The algorithm terminates after $\mathtt{diam}$ rounds. The agents that still have $\mathtt{d_i} = \hat{d}_i$ at the end know that they belong to $\arg\min_{i\in[n]} \hat{d}_i$. This algorithm can also be implemented in an asynchronous network by adding round numbers to the transmitted messages [36, section 15.2].

III. Adaptive Coverage Control Algorithms

A. Coverage Control for Mobile Robotic Networks

In the standard coverage control problem [4], the goal of the robotic network is to reach asymptotically a configuration where the agent positions $\lim_{t\to\infty} p_i(t), i \in [n]$, minimize the following performance measure capturing the quality of coverage of certain events:

$$\mathcal{E}_n(p) = \mathbb{E}_z \left[\min_{i\in[n]} f(\|p_i - Z\|) \right], \tag{4}$$

where $f : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is an increasing continuously differentiable function. Here $\mathbb{E}_z$ is the expectation operator corresponding to the probability distribution $\mathbb{P}_z$ of the random variable Z, and we assume that the support of $\mathbb{P}_z$ is contained in the workspace Q. The value $\mathbb{P}_z(\mathsf{A})$ represents the probability of an event appearing in the set $\mathsf{A} \subset \mathsf{Q}$. An event must be serviced by the robot closest to the location of this event. The cost of servicing an event at location z with a robot at location p_i is is measured by $f(\|p_i - z\|)$. For example, in vehicle routing problems, this cost can be the time it takes a robot to travel to the event location, i.e., $f(\|p_i - z\|) = \|p_i - z\|/v_i$, see Section V. In sensing scenarios, $f(\|p_i - z\|)$ measures the degradation of the sensing performance with the distance to the event [4]. Depending on the application, events are alternatively called jobs, demands, or targets. The robots are also called agents or vehicles. Note that the objective (4) assumes that every time an event occurs, the robotic network is in its desired configuration. Hence if servicing the events requires that the robots move, e.g. to the event location as in vehicle routing problems [14], the coverage control framework is only applicable in light load conditions where enough time separates successive events to let the robots return to their reference position $p_i, i \in [n]$.

In [4] and most of the subsequent literature, it is assumed that the event distribution $\mathbb{P}_z$ is known. The network can then reach its desired configuration *before any event occurs*, and the minimization of (4) is essentially an open-loop optimization problem. Indeed with $\mathbb{P}_z$ known one can implement a gradient descent algorithm to locally minimize the objective (4). Assuming for simplicity that the agents are synchronized, and a constant sampling period $T > 0$, we denote the agents positions at time kT by $p_k := p(kT) = [p_{1,k}^T, \dots, p_{n,k}^T]^T$. The robots start at $p_0 = [p_{1,0}, \dots, p_{n,0}]$ at $t = 0$ and update their positions according to

$$p_{i,k+1} = p_{i,k} - \gamma_k \frac{\partial \mathcal{E}_n}{\partial p_i}\Big|_{p_k}, \tag{5}$$

where γ_k is an appropriately chosen sequence of decreasing or small constant stepsizes. Throughout the paper $\partial \mathcal{E}_n / \partial p_i$ for $p_i \in \mathbb{R}^q$ denotes the q-dimensional vector of partial derivatives with respect to the components of p_i. Minor modifications might be required to accommodate velocity constraints in (5) and are discussed in subsection III-C. The agents implementing (5) then asymptotically reach a configuration that is a critical point of $\mathcal{E}_n$. No guarantee to reach a global minimum is offered in general, and indeed global minimization of the function (4) can be difficult [37]. Nevertheless, an interesting property of the gradient descent algorithm (5) for the coverage control problem is that it can be implemented in a distributed manner by the robots, by exploiting the following result.

Proposition 1. *Assume that hyperplanes in $\mathbb{R}^q$ have $\mathbb{P}_z$-measure zero. Then $\mathcal{E}_n$ is globally Lipschitz on Q^n, and continuously differentiable on $\mathsf{Q}^n \setminus \mathsf{D}_n$, with partial derivatives*

$$\frac{\partial \mathcal{E}_n}{\partial p_i}\Big|_p = \int_{V_i(p)} f'(\|p_i - z\|) \frac{p_i - z}{\|p_i - z\|} \mathbb{P}_z(dz). \tag{6}$$

Here we adopt the convention $0/\|0\| := 0$.

Remark 1. Note that the assumption that hyperplanes in have $\mathbb{P}_z$-measure zero implies that points also have measure zero, and so in particular the support of $\mathbb{P}_z$ is infinite.

The proof of this proposition can be found in [38, Proposition 9], [39]. We then see that each agent can update its position at each period according to (5) by communicating only with its current Voronoi neighbors, in order to determine the boundaries of its own Voronoi cell $V_i(p)$ and compute the integral (6). Even in a large network, a single robot has typically only few Voronoi neighbors, which allows for a scalable and distributed implementation of the gradient descent algorithm.

Remark 2. The specific case where $f(x) = x^2$ is considered for coverage control in [4] in more detail. In this case (6) gives

$$\frac{\partial \mathcal{E}_n}{\partial p_i}|_{p=p_k} = 2\mathbb{P}_z(V_i(p_k))p_{i,k} - \int_{\mathsf{V}_i(p_k)} z\mathbb{P}_z(dz). \tag{7}$$

Assuming that $\mathbb{P}_z(V_i(p_k)) \neq 0$, define the centroid of the Voronoi region $V_i(p_k)$ as

$$C_{V_i(p_k)} = \frac{1}{\mathbb{P}_z(V_i(p_k))} \int_{\mathsf{V}_i(p_k)} z\mathbb{P}_z(dz).$$

Then control law (5), i.e.,

$$p_{i,k+1} = p_{i,k} - \gamma_k \frac{\partial \mathcal{E}_n}{\partial p_i}\Big|_{p_k} = p_{i,k} - 2\gamma_k \mathbb{P}_z(V_i(p_k))(p_{i,k} - C_{V_i(p_k)}),$$

is essentially the well-known Lloyd least-squares quantization algorithm [7].

A limitation of the gradient descent algorithm (5) for coverage control is that it does not include any feedback mechanism that would exploit actual observations of the successive event locations to correct for potential modeling errors in the assumed target distribution $\mathbb{P}_z$. Moreover, our goal in this paper is to develop deployment algorithms that work with an unknown event distribution $\mathbb{P}_z$, in which case the updates (5) simply cannot be computed. We allow instead the robots to update their positions based on the observed successive event locations. The main idea for our approach, based on using stochastic gradient algorithms rather than the deterministic algorithm (5), is described in the next subsection. Subsection III-C applies this idea to the adaptive coverage control problem.

B. *Stochastic Gradient Algorithms*

Assume that we wish to minimize a function F of the form

$$F(x) = \mathbb{E}_z[f(x, Z)] = \int f(x, z) d\mathbb{P}_z(z), \tag{8}$$

such as $\mathcal{E}_n$ defined in (4) for example. Contrary to the previous subsection, we now assume that $\mathbb{P}_z$ is unknown, so that the expectation cannot be computed directly. Let us assume that f is differentiable with respect to x, for $\mathbb{P}_z$-almost all z, and denote its gradient $\nabla_x f(x, Z) := \frac{\partial f(x,Z)}{\partial x}$. Finally, assume that we can observe random variables $Z_k, k \geq 1$, iid with distribution $\mathbb{P}_z$. Consider then the recursive algorithm

$$x_{k+1} = x_k - \gamma_k \nabla_x f(x_k, Z_{k+1}), \tag{9}$$

which can be rewritten in the form

$$x_{k+1} = x_k + \gamma_k (h(x_k) + D_{k+1}), \tag{10}$$

with $h(x_k) = -\mathbb{E}[\nabla_x f(x_k, Z_{k+1})|x_k]$ and $D_{k+1} = -\nabla_x f(x_k, Z_{k+1}) + \mathbb{E}[\nabla_x f(x_k, Z_{k+1})|x_k]$. Note that for Z_{k+1} a random variable, $\nabla_x f(x_k, Z_{k+1})$ is a random vector, called a stochastic gradient of f. Define the increasing family of σ-algebras $\mathcal{F}_k := \sigma(x_0, D_i, 1 \leq i \leq k)$. Then $\{D_k\}_{k \geq 1}$ is a martingale difference sequence with respect to $\mathcal{F}_k$, i.e. $E[D_{k+1}|\mathcal{F}_k] = 0, \ \forall k \geq 0$. Under broad conditions and with an appropriate choice of stepsizes γ_k, the ODE method [40] says that asymptotically the sequence $\{x_k\}_{k \geq 0}$ in (10) almost surely approaches the trajectories of the ODE

$$\dot{x} = h(x). \tag{11}$$

Classical almost sure convergence results are obtained under the condition

$$\sum_{k=0}^{\infty} \gamma_k = +\infty, \quad \sum_{k=0}^{\infty} \gamma_k^2 < +\infty,$$

which holds for $\gamma_k = 1/(1+k)$ for example. In many applications however, the stepsizes γ_k are chosen to converge to a small positive constant, which allows tracking of the equilibria of (11) if the problem parameters (e.g. $\mathbb{P}_z$) change with time. In this case, one typically obtains convergence to a neighborhood of an equilibrium of (11). The selection of proper stepsizes is an important practical issue that is not emphasized in this paper but is discussed at length in references on stochastic approximation algorithms [41], [42].

Assuming that it is valid to interchange expectation and derivation in the definition of h, we have

$$h(x) = -\mathbb{E}[\nabla_x f(x, Z)|x] = -\nabla F(x). \tag{12}$$

Note that proposition 1 precisely says that the differentiation under the integral sign formula (12) is valid for $\mathcal{E}_n$ under the assumption that hyperplanes have $\mathbb{P}_z$-measure zero. In this case, the iterates (9) asymptotically approach the limit set of the gradient flow $\dot{x} = -\nabla F(x)$, which are the critical points of F. In general we can in fact expect convergence to the set of local minima of F. This device allows us to reach these minima in the absence of knowledge of $\mathbb{P}_z$, as long as we have access to realizations of the random variables Z_k.

C. Adaptive Coverage Control

We now consider the following modification of the coverage control problem. The events appear randomly in the workspace, with event k appearing at time $t_k > 0$ and location $Z_k \in \mathsf{Q}$, for $k \geq 1$. We let $t_0 := 0$ denote the initial time. Assume in this subsection that the successive locations of the events $Z_k, k \geq 1$, are iid with probability distribution $\mathbb{P}_z$ on Q. The distribution $\mathbb{P}_z$ is now unknown, and hence the deterministic gradient descent algorithm (5) cannot be computed. We assume that hyperplanes in $\mathbb{R}^q$ have $\mathbb{P}_z$-measure zero, so that the gradient formula (6) holds.

We denote the agent positions at time t_k^-, i.e., right before the occurrence of the k^{th} event, by $p_{k-1} = [p_{1,k-1}^T, \ldots, p_{n,k-1}^T]^T \in (\mathbb{R}^q)^n$, for $k \geq 1$. These positions are called *reference positions* and are updated according to

$$p_{i,k+1} = p_{i,k} + u_{i,k}, \quad |u_{i,k}| \leq v_{i,k}, \; \forall k \in \mathbb{Z}_{\geq 0}, \forall i \in [n], \tag{13}$$

where $u_{i,k} \in \mathbb{R}^q$ is a control input for the interval $[t_k, t_{t+1})$. For example, if the robot dynamics follow the model (1) and servicing the targets requires no additional travel, we can take $v_{i,k} = v_i(t_{k+1} - t_k)$ for all $i \in [n]$. We assume that there exists a constant $v > 0$ such that $v_{i,k} \geq v$ for all $i \in [n]$ and $k \geq 0$, so that the robots can update their reference positions by a non-vanishing positive distance at each period.

When the k^{th} event occurs at time t_k and position $Z_k \in \mathsf{Q}$, $k \geq 1$, we assume that at least the robot closest to that event location can observe it. This robot, say robot i, services the target starting from its location $p_{i,k-1}$, and then moves to its new reference position $p_{i,k}$. Using the result of Proposition 1, assuming the hyperplanes in $\mathbb{R}^q$ have $\mathbb{P}_z$-measure zero, and ignoring for now the velocity constraints $v_{i,k}$, it is easy to see that the following reference position updates implement the stochastic gradient algorithm (9) to minimize the coverage objective (4)

$$p_{i,k+1} = \begin{cases} p_{i,k} + \gamma_k f'(\|p_{i,k} - Z_{k+1}\|) \frac{Z_{k+1} - p_{i,k}}{\|Z_{k+1} - p_{i,k}\|} & \text{if robot } i \text{ is closest to } Z_k, \\ p_{i,k} & \text{otherwise.} \end{cases} \tag{14}$$

Indeed the quantity $f'(\|p_{i,k} - Z_{k+1}\|) \frac{Z_{k+1} - p_{i,k}}{\|Z_{k+1} - p_{i,k}\|} \mathbf{1}_{\{Z_{k+1} \in V_i(p_k)\}}$ is an unbiased estimate of the gradient (6). The determination of the closest robot to the target in the first case of (14) can be done in a distributed way via the `FloodMin` algorithm described in paragraph II-D, with the agents initializing their value to $\hat{d}_i = \|p_{i,k} - Z_{k+1}\|$ if they detect the event, and to $\hat{d}_i = +\infty$ if they are too far away to detect it. If several agents find that they are the closest to the target, which happens with probability zero under our assumption that hyperplanes have $\mathbb{P}_z$-measure zero, we can either implement a mechanism to resolve the ties arbitrarily or let all these agents change their reference position. Clearly there are other ways, depending on the scenario, to implement rule (14). For example, in the context of the DTRP, we could let all the robots travel to the event location at the same speed, as in [23], a scheme that does not require any coordination. Then only the first robot to reach the target changes its reference position for the next period.

We can modify update law (14) slightly, in order to account for the motion constraint $v_{i,k}$ and to avoid the situation where a robot following (14) lands outside of the workspace Q (this can happen for certain functions such as $f(x) = x$). Define, for a vector $u \in \mathbb{R}^q$ and a scalar $b > 0$, the truncation $[\text{sat}(u)]_b$ by

$$[\text{sat}(u)]_b = \begin{cases} u, & \text{if } \|u\| \leq b, \\ b \frac{u}{\|u\|}, & \text{if } \|u\| > b. \end{cases}$$

Then consider the modified update rule, compatible with (13)

$$p_{i,k+1} = \begin{cases} \Pi_{\mathsf{Q}} \left[p_{i,k} + \text{sat} \left[\gamma_k f'(\|p_{i,k} - Z_{k+1}\|) \frac{Z_{k+1} - p_{i,k}}{\|Z_{k+1} - p_{i,k}\|} \right]_{v_{i,k}} \right] & \text{if robot } i \text{ is closest to } Z_{k+1}, \\ p_{i,k} & \text{otherwise,} \end{cases} \tag{15}$$

where Π_{Q} is the orthogonal projection on the convex set Q.

It is interesting to note that the stochastic gradient descent update (14) or (15) is typically much easier to compute than the corresponding deterministic gradient update based on (6). No Voronoi cell computation or integration is required, only a distributed mechanism to find which robot is the closest to the target. We also note that this procedure could in fact also be used in the situation where $\mathbb{P}_z$ is known, by generating random targets artificially, essentially evaluating the integral (6) by Monte-Carlo simulation. This approach is typically not competitive with the deterministic integration methods for small values of the dimension q and simple distributions $\mathbb{P}_z$ and functions f, but still useful in general [38].

Special Cases: If we specialize (14) to the least-squares coverage problem with $f(x) = x^2$, we obtain the update $p_{i,k+1} = p_{i,k} + \gamma_k(Z_{k+1} - p_{i,k})$ for the closest robot. This particular adaptive algorithm has been used extensively in various fields, from statistics to quantization to neural networks [5], [27], [28]. If $f(x) = x$ and all robots travel at unit speed, the service cost for an event appearing at Z_k is the time it takes for the closest robot to travel to the event location. In this case, the update rule (15) is simply $p_{i,k+1} = p_{i,k} + \gamma_k \frac{Z_{k+1} - p_{i,k}}{\|Z_{k+1} - p_{i,k}\|}$ for the closest robot. It provides a simpler solution to the adaptive DTRP in light load considered recently by Arsie et al. [23]. In contrast, their algorithm requires the vehicles to keep track of all the past events they serviced so far and to compute a median of this growing list at each iteration. The DTRP is discussed in more details in section V.

Remark 3. For certain distributions and initial robot positions outside of the support set of the distribution, it is possible that by following (15), some agents will never move. The issue also arises in the deterministic case however, since if $\mathbb{P}_z(V_i(p_k)) = 0$ then the gradient (7) vanishes. A possible solution to avoid this phenomenon is to add an initial transient regime where for example all agents follow the first case of the rule (15) rather than simply the closest agent. The goal of this transient modification is thus to bring all the robots within the support set of the target distribution. It is either stopped at some finite time or discounted by a stepsize decreasing much faster that γ_k, thereby not impacting the convergence results.

We now state a convergence result for the update law (15) to the set of critical points of the objective $\mathcal{E}_n$, i.e., to

$$\mathcal{H}_n^{\text{ode}} = \{x \in \mathsf{Q}^n \setminus \mathsf{D}_n | \nabla \mathcal{E}_n(x) = 0\}. \tag{16}$$

Even though the algorithm is a stochastic gradient algorithm, the discontinuity of $\nabla \mathcal{E}_n$ on the set D_n creates technical difficulties. To the best of our knowledge, the most thorough investigation of the dynamics of (14) can be found in [38] and leaves open the question of non-convergence to D_n. Our strategy differs somewhat from that paper. We cope with the non-differentiability on

D_n by introducing the the Fillipov set-valued map

$$F(p) = \bigcap_{\delta>0} \mathrm{co}\left(\bigcup_{\hat{p}\in B(p,\delta)} \nabla\mathcal{E}_n(\hat{p})\right), \tag{17}$$

where co denotes the convex hull. Then for $p \notin \mathsf{D}_n, F(p) = \{\nabla\mathcal{E}_n(p)\}$ is a single-valued map because $\mathcal{E}_n$ is continuously differentiable at p by Proposition 1 [43]. For $p \in \mathsf{D}_n$, $F(p)$ is the set of all convex combinations of the gradient vectors that can be obtained as limits when some of the robots converge to the same position [43]

$$\forall p \in \mathsf{D}_n, \quad F(p) = \mathrm{co}\left\{\lim_{k\to\infty} \nabla\mathcal{E}_n(p_k) | p_k \to p \text{ as } k \to \infty\right\}.$$

Theorem 1. *Assume that* $\sum_{k\geq 0} \gamma_k = +\infty$*,* $\sum_{k\geq 0} \gamma_k^2 < \infty$*,* $p_0 \in \mathsf{Q}^n \setminus \mathsf{D}_n$*, and that hyperplanes in* $\mathbb{R}^q$ *have* $\mathbb{P}_z$*-measure* 0*. Then by following the updates (14) or (15), the sequence* $\{p_k\}_{k\geq 1}$ *of robot positions converge almost surely to a compact connected subset of* $\mathcal{H}_n^{ode} \cup \mathcal{H}'$*, invariant for the differential inclusion* $\dot{x} \in F(x)$*, where* $\mathcal{H}' \subset \mathsf{D}_n$*.*

If in addition $\mathbb{P}_z$ *dominates the Lebesgue measure on* Q*, then the robot positions converge almost surely to a compact connected subset of* $\mathcal{H}_n^{ode}$*. Hence if* $\mathcal{E}_n$ *has only isolated critical points in* $\mathsf{Q}^n \setminus \mathsf{D}_n$*, the sequence* $\{p_k\}_{k\geq 0}$ *converges to one of them almost surely.*

The proof of Theorem 1 can be found in appendix A. Note that in the first part of the theorem, we do not rule out the convergence to equilibria of the differential inclusion $\dot{x} \in F(x)$ situated on the set D_n of aggregated configurations. These equilibria are in fact critical points of $\mathcal{E}_k$ for $k < n$, with several agents occupying the same position. It is reasonable to conjecture that such asymptotic aggregated formations do not in fact occur, at least if the event distribution is "sufficiently rich", and this motivates the second part of the theorem, although we do not claim to provide the most general result. Note that almost-surely the update rule (14) or (15) never results in two robots landing on the same position as long as $q \geq 2$, because this would require Z_{k+1} to fall on a line containing these two robot positions. Hence almost surely $p_k \notin \mathsf{D}_n$ for any finite k. This can be achieved for $q = 1$ as well by a slight perturbation of the sequence γ_k subject to the conditions of Theorem 1 being satisfied. The second part of the theorem also rules out asymptotic convergence of $\{p_k\}_{k\geq 0}$ to D_n.

Remark 4. The analysis above extends immediately to the case where $v_{i,k} \geq v > 0$ for all $i \in [n]$ for infinitely many $k \geq 0$, by not updating the reference positions during the periods where this condition in not met, and renumbering the periods to account only for those where the condition is met.

D. Some Extensions

Before closing this section, we briefly illustrate how stochastic gradient algorithms provide simple solutions to interesting variations on the coverage control problem.

1) Target Tracking with Markovian Dynamics: In subsection III-C, we assumed that the successive locations Z_k were iid. Instead, let us assume now that we wish to track a single target in discrete time, whose position at time t_k is Z_k, where Z_k evolves as an ergodic Markov chain with stationary distribution $\mathbb{P}_z$. The objective is still to optimize $\mathcal{E}_n$ defined by (4), which represents the steady-state tracking error. We can then use algorithm (15) to optimize the robotic network configuration, and the convergence result of Theorem 1 is still valid. This tracking scheme does not require knowledge of the target dynamics nor that of the stationary distribution $\mathbb{P}_z$.

As an example, consider a target moving on a circle of radius R, with dynamics

$$\theta_{k+1} = 0.95\,\theta_k + \xi_k,$$

where the variables ξ_k are iid uniform on $[-0.5, 0.5]$ and $Z_k = [R\cos\theta_k, R\sin\theta_k]^T$. The result of the adaptive coverage algorithm for $f(x) = x^2$ is shown on Fig. 1. Note that the target distribution clearly does not dominate the Lebesgue measure as required in the second part of theorem 1, yet in practice we do not observe convergence to an aggregated configuration. The robots aggregate in the region around the point $[1, 0]^T$ where the target spends most of its time.

2) A Heterogenous Coverage Problem: As in subsection III-C, an event appears randomly in the environment at each period and must be serviced. However, let us now assume that there are two types of agents, with m_A robots of type A and m_B robots of type B, and three types of events a, b, ab. Events of type a must be serviced by a robot of type A, events of type B by a robot of type b, and events of type ab by a robot of type A *and* a robot of type B. When a new event appears, it is of type α with some unknown probability λ_α, $\alpha \in \{a, b, ab\}$, and the agents can observe its type. The spatial distribution of events of type α is $\mathbb{P}_\alpha$ and is a also a priori unknown. The corresponding expectation operator is denoted $\mathbb{E}_\alpha$. Finally, denote the vector of robot positions $p = [p_1^A, \ldots, p_{m_A}^A, p_1^B, \ldots, p_{m_B}^B]$. Assume that the asymptotic configuration of the

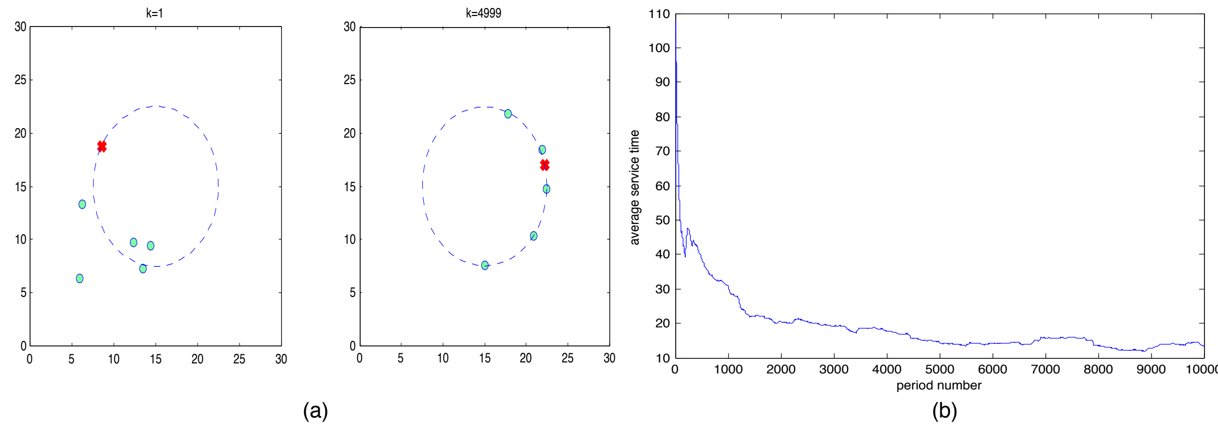

Fig. 1. Adaptive coverage algorithm for a target with Markovian dynamics moving on a circle. We show the initial configuration of the robots (blue circles) and the target (red cross) and the configuration after 5000 time-steps. The stepsizes used were $\gamma_k = 1/(1+5\times10^{-3}k)$. The curve on the right shows the evolution of the empirical average cost over time, where the average is taken over the past 1000 cost measurements.

robots must now optimize the expected cost

$$\begin{aligned}\mathcal{E}_{m_A,m_B}(p) =& \lambda_a \mathbb{E}_a\left[\min_{i\in[m_A]} f_A(\|p_i^A - z\|)\right] + \lambda_b \mathbb{E}_b\left[\min_{j\in[m_B]} f_B(\|p_j^B - z\|)\right] \\ &+ \lambda_{ab}\mathbb{E}_{ab}\left[\min_{\substack{i\in[m_A]\\ j\in[m_B]}}\left\{\max\left\{f_A(\|p_i^A - z\|), f_B(\|p_j^B - z\|)\right\}\right\}\right],\end{aligned} \tag{18}$$

where f_A and f_B are increasing, continuously differentiable functions. Note that the cost of servicing an event of type ab is the maximum of the costs of servicing it with one robot of each type.

For this problem, one can verify that the stochastic gradient update rule takes the following surprisingly simple form [39]. When an event of type a appears at z_{k+1}, the closest robot of type A, say i, services it and changes it reference position by moving it toward z_{k+1} by a (possibly truncated) step $\gamma_k f_A'(\|z_{k+1} - p_{i,k}^A\|)\frac{z_{k+1}-p_{i,k}^A}{\|z_{k+1}-p_{i,k}^A\|}$, and similarly for a target of type b and a robot of type B. If the target is of type ab, the closest A and B robots service it. To update their reference positions for the next period, they first find which of the two is the *farthest* from the event. Then only this robot moves its reference position by the same step. In view of the complicated expression of the objective function, such a simple rule based update law is quite appealing. We illustrate its behavior on Fig. 2 for $f_A(x) = f_B(x) = x$. In regions where events

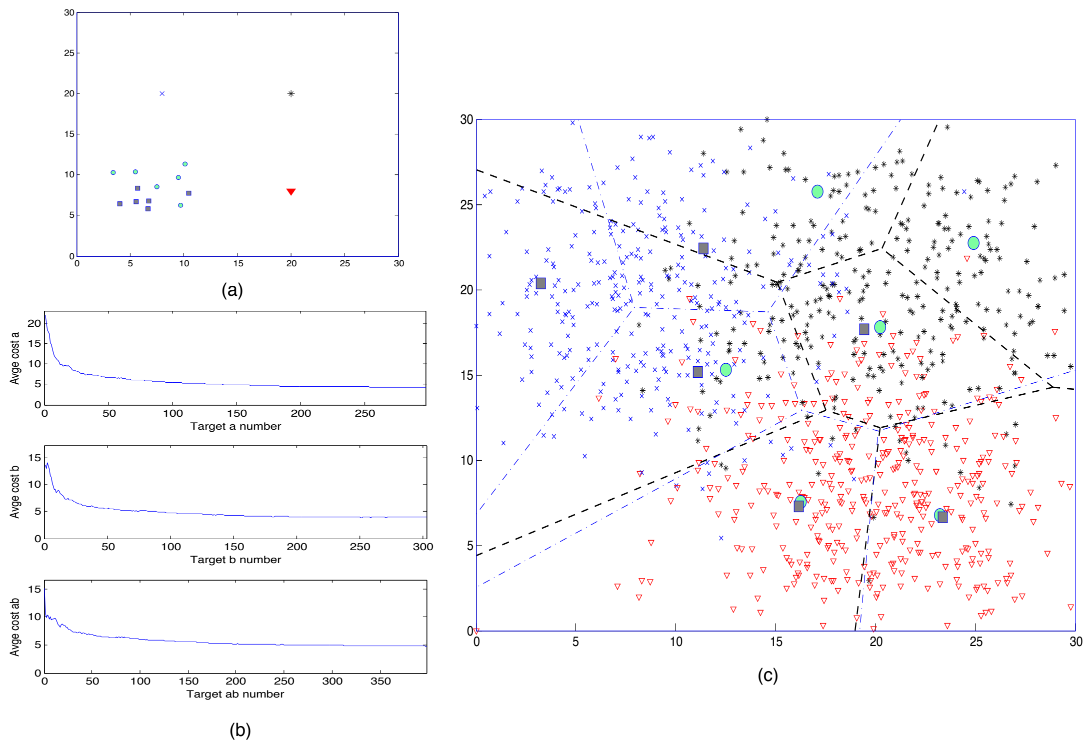

Fig. 2. Heterogeneous coverage control for a system with two types of robots, A (green circles) and B (gray squares). Events requiring service from type a appear with probability 30% and a distribution approximately centered at $[20; 20]^T$ (star on Fig. (a)). Targets of type b appear with probability 30% and a distribution approximately centered at $[8; 20]^T$ (cross on Fig. (a)). Finally targets of type ab appear with probability 40% and a distribution approximately centered at $[20; 8]^T$ (triangle on Fig. (a)). Fig. (a) shows the initial robot configuration and Fig. (c) the configuration reached after 1000 targets, together with the history of target locations. The Voronoi cells of each robot are indicated but not computed by the algorithm (separate Voronoi diagrams are drawn for the two robot types). Note how robots of type A and B tend to pair in the lower right corner in order to service the targets of type ab efficiently (here $f_A(x) = f_B(x) = x$). Fig. (b) shows the empirical average cost incurred by the targets of each type, where the average is taken over all the past targets of the same type seen so far.

of types ab appear most frequently, we observe in general that a pair of robots of different types will aggregate toward the same position in order to service these events efficiently. Of course, this does not necessarily happen if other events of type a or b can appear in the region, since each robot must then balance its requirements to serve two types of events.

IV. Adaptive Spatial Load-Balancing and Partitioning

In this section, we design distributed adaptive algorithms that partition the workspace Q into n cells, one for each robot, so that the steady-state probability that an event falls in cell i

has a prespecified value a_i. Here we have $a_i \geq 0, i \in [n]$, and $\sum_{i=1}^n a_i = 1$. By letting each robot service only the events occurring in his cell, these algorithms allow us to specify the steady state utilization of the different agents. Such spatial load balancing algorithms have important applications in multi-robot systems and location optimization, see e.g. [20], [21], [29]. An application to the DTRP is described in Section V.

As in Section III-C, events occur at times t_k and iid locations $Z_k, k \geq 1$, and the unknown distribution $\mathbb{P}_z$ has support included in Q. Based on the observation of the successive event locations, we design a sequence of partitions of Q into regions $\{R_{i,k}\}_{i\in[n]}$, $k \geq 0$, such that at period $k \geq 1$, agent i is responsible for servicing the event if and only if $Z_k \in R_{i,k-1}$. Here we slightly abuse terminology and allow our partitions to have $R_{i,k} \cap R_{j,k} \neq \emptyset$ for $i \neq j$. Then if Z_k falls in the intersection of several regions, any of the corresponding agents can service the event. Our algorithms produce regions whose intersections have $\mathbb{P}_z$-measure zero, hence this case has no influence on the final result. After the k^{th} event occurs, the agents can change the boundaries of their respective regions to form the partition $\{R_{i,k}\}_{i\in[n]}$ used to decide which agent services the $(k+1)^{th}$ event.

Our sequence of partitions $\{R_{i,k}\}_{i\in[n]}$ converges to a partition $\{R_i\}_{i\in[n]}$, i.e., $d_H(R_{i,k}, R_i) \to 0$ as $k \to \infty$, such that $\mathbb{P}_z(R_i) = a_i$ for all $i \in [n]$. Let $\mathcal{G} = \{g_1, \dots, g_n\}$ be n fixed points in $\mathbb{R}^q$, with point g_i associated to robot i. We call the point g_i the generator of region R_i. Designing a partition $\{R_i\}_{i\in[n]}$ is equivalent to choosing an *assignment* of event locations to region generators, i.e., a measurable map $T : \mathsf{Q} \to \mathcal{G}$, by taking $R_i = T^{-1}(g_i), i \in [n]$. Let us denote the set of all such assignments by $\mathcal{T}$. We then look for an assignment $T \in \mathcal{T}$ satisfying the constraint $\mathbb{P}_z(T^{-1}(g_i)) = a_i, i \in [n]$, and design recursive algorithms producing such an assignment asymptotically.

There are many ways of designing such regions or assignments. In particular, consider the following optimization problem

$$\inf_{T\in\mathcal{T}} \quad \int_{\mathsf{Q}} c(z, T(z))\mathbb{P}_z(dz) \tag{19}$$

$$\text{subject to} \quad \mathbb{P}_z(T^{-1}(g_i)) = a_i, \;\; i \in [n], \tag{20}$$

where $c : \mathsf{Q} \times \mathcal{G} \to \mathbb{R}$ is a given cost function. For $w \in \mathbb{R}^n$ a parameter, define by analogy with

(3) the generalized Voronoi regions

$$\tilde{V}_i^c(\mathcal{G}, w) := \{z \in \mathsf{Q} | c(z, g_i) - w_i \le c(z, g_j) - w_j,\ j \ne i\},\ \forall i \in [n].$$

The following theorem generalizes some results in [20], [21], [29] by imposing weaker conditions on $\mathbb{P}_z$ and c. A proof is provided in appendix B, based on results from optimal transportation [32]–[34]. To give an indication of the generality of the possible results [32], we also remove our assumptions on Q from section II-B.

Theorem 2. *Consider problem (19), (20), where $(\mathsf{Q}, \mathbb{P}_z)$ is a probability space with Q, and assume that*

A1) For all $i \in [n]$, $z \to c(z, g_i)$ is lower semi-continuous on Q and $z \to \max_{i \in [n]} c(z, g_i)$ is $\mathbb{P}_z$-integrable.

A2) For all $i \ne j \in [n]$, for all $r \in \mathbb{R}$, the set $\{z \in \mathsf{Q} : c(z, g_i) - c(z, g_j) = r\}$ has $\mathbb{P}_z$-measure zero.

Then the problem admits an assignment $T \in \mathcal{T}$ that attains the infimum in (19). The value of the optimization problem is equal to

$$\max_{w \in \mathbb{R}^n} h(w) := \int_{\mathsf{Q}} \min_{i \in [n]} \{c(x, g_i) - w_i\} \, \mathbb{P}_z(dz) + \sum_{i=1}^n a_i w_i, \tag{21}$$

and this maximum is attained for some $w^* \in \mathbb{R}^n$. An optimal assignment T is then given by the generalized Voronoi regions

$$\forall z \in \mathsf{Q},\ \ T(z) = g_i \Leftrightarrow z \in \tilde{V}_i^c(\mathcal{G}, w^*).$$

Finally, h is a concave function, and a supergradient of h at w is given by

$$[-\mathbb{P}(\tilde{V}_1^c(\mathcal{G}, w)) + a_1, \dots, -\mathbb{P}(\tilde{V}_n^c(\mathcal{G}, w)) + a_n]^T. \tag{22}$$

Hence the following supergradient optimization algorithm

$$\begin{aligned} & w_0 = 0, \\ & w_{i,k+1} = w_{i,k} + \gamma_k[-\mathbb{P}(\tilde{V}_i^c(\mathcal{G}, w_k)) + a_i],\ i = 1, \dots, N, \end{aligned} \tag{23}$$

where γ_k is a sequence of positive stepsizes decreasing to 0 such that $\sum_{k=0}^\infty \gamma_k = +\infty, \sum_{k=0}^\infty \gamma_k^2 < \infty$, converges to an optimal set of weights maximizing h.

In other words, there is a set of weights $w^* \in \mathbb{R}^n$, maximizing of the dual function defined in (21), for which the corresponding generalized Voronoi cells $\{\tilde{V}_i^c(\mathcal{G}, w^*)\}_{i\in[n]}$ satisfy the constraint of interest (20). In addition, the assignment corresponding to these regions minimizes (19). In practice, we make additional assumptions on the function c to obtain reasonably shaped regions. In particular, if $c(z, g_i) = \|z - g_i\|^2$, then the abstract Voronoi diagrams become power diagrams. Because the boundaries of the power cells are hyperplanes in $\mathbb{R}^q$ [25], our assumption A2 on $\mathbb{P}_z$ in Theorem 2 is satisfied if hyperplanes have $\mathbb{P}_z$-measure zero, as in Section III.

In our case, since $\mathbb{P}_z$ is unknown, we replace the supergradient (22) by a stochastic supergradient. Let us specialize the discussion to $c(z, g_i) = f(\|z - g_i\|)$, where f is increasing. In this case we have denoted the generalized Voronoi cells in (3) by $V_i^f(\mathcal{G}, w)$. If, at period k, the event is located at Z_k, a possible choice for this stochastic supergradient is simply

$$[-\mathbf{1}_{\{V_1^f(\mathcal{G},w_{k-1})\}}(Z_k) + a_1, \ldots, -\mathbf{1}_{\{V_n^f(\mathcal{G},w_{k-1})\}}(Z_k) + a_n]^T. \tag{24}$$

Note that it is much easier to test if $Z_k \in V_i^f(g, w_{k-1})$ than to compute the generalized Voronoi cell, and this is all that is required to compute (24). Assuming that at least the robot associated with the region $R_{i,k-1}$ where the k^{th} event occurs detects the event, the agents can simply run the `FloodMin` algorithm (see subsection II-D) with $\hat{d}_i = f(\|Z_k - g_i\|) - w_{i,k}$ (and $\hat{d}_i = +\infty$ if agent i did not detect the event).

Algorithm 1 Adaptive partitioning algorithm

Require: for robot i: its desired utilization rate a_i, and the function f such that $c(z, g_i) = f(\|z - g_i\|)$ in (19).

Robot i initializes its weight to $w_i = 0$, $i \in [n]$.

When the k^{th} new event appears at location Z_k, for $k \geq 1$:

Run the `FloodMin` algorithm with $\hat{d}_j = f(\|Z_k - g_j\|) - w_j, j \in [n]$.

If robot i has $d_i = \hat{d}_i$, it updates its weight as $w_i \leftarrow w_i + \gamma_{k-1}(a_i - 1)$

Otherwise, it updates its weight as $w_i \leftarrow w_i + \gamma_{k-1} a_i$.

Algorithm 1 is then a stochastic supergradient algorithm computing the optimal weights of the generalized Voronoi partition, and asymptotically this partition satisfies the constraints (20) almost surely. The behavior of this algorithm is illustrated on Fig. 3. The following theorem is now a direct application of well known convergence results for stochastic subgradient algorithms, see e.g. [41].

Theorem 3. *Choose the stepsizes γ_k in algorithm 1 so that $\sum_{k=0}^{\infty} \gamma_k = \infty$, $\sum_{k=0}^{\infty} \gamma_k < \infty$. Assume that condition A2 of Theorem 2 is satisfied for $c(z, g_i) = f(\|z - g_i\|)$. Then almost surely, the weights updated following algorithm 1 converge to a maximizer w^* of (21), and the resulting generalized Voronoi diagram $\{V_i^f(\mathcal{G}, w^*)\}_{i \in [n]}$ satisfies the utilization constraints (20).*

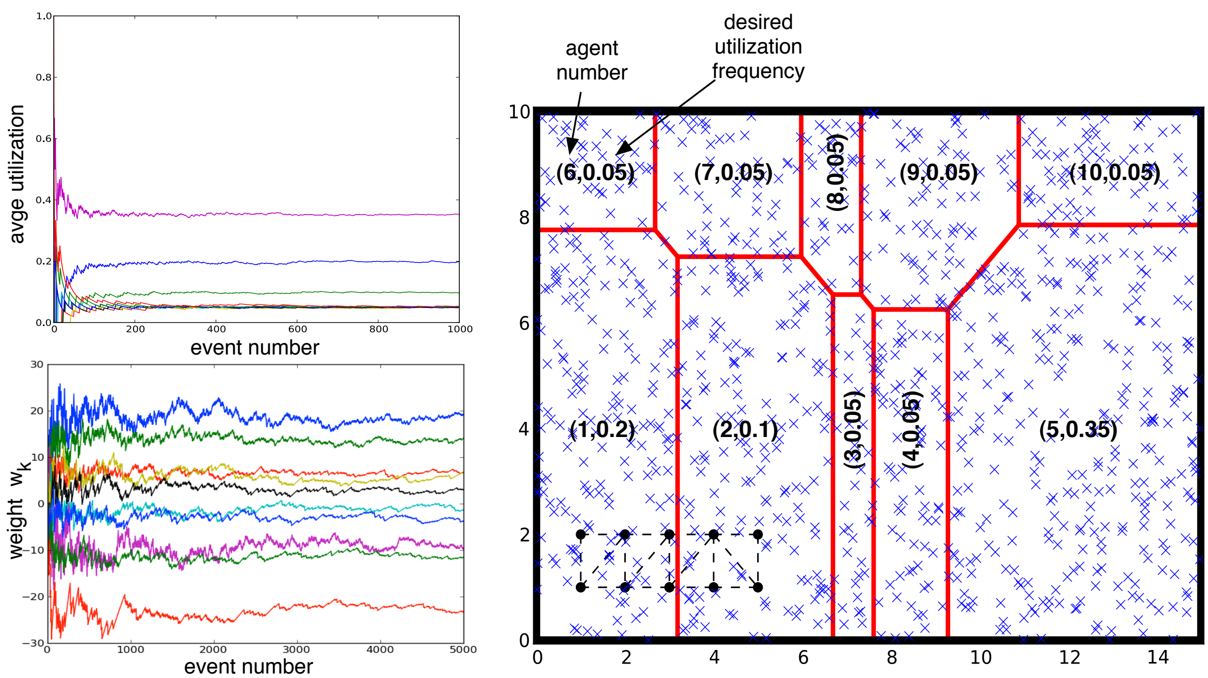

Fig. 3. Partition for 10 robots after 1000 events for the quadratic cost $c(z, g_i) = \|z - g_i\|^2$. The partition at each step is a power diagram. The desired utilization rates are shown for each agent on the figure. The power diagram generators used are represented as black dots in the lower left corner. Note that fixing their positions determines the directions of the cell boundaries. The power cells shown in red are computed using CGAL [44], but need not be computed by the agents running the stochastic gradient algorithm. The top left figure shows the evolution of the empirical utilization frequencies over the first 1000 events, and the top right figure the evolution of the weight vector w_k. The chosen stepsizes were $\gamma_k = 10/(1 + 0.01k)$.

V. AN ADAPTIVE DYNAMIC VEHICLE ROUTING ALGORITHM

We now combine the algorithms of Section III-C and Section IV to design an adaptive algorithm for the Dynamic Traveling Repairman Problem (DTRP). Assume for simplicity in this section that the environment is planar, i.e., $q = 2$. The DTRP was initially studied in [14], [16], and more recently in e.g. [17]. In these references, the proposed algorithms require the knowledge of the event distribution. The recent references [19], [23] propose algorithms for the DTRP that work without knowledge of $\mathbb{P}_z$ in the light traffic regime, but left open the adaptive problem in heavy traffic. We solve this open problem using the adaptive partitioning algorithm of section IV. In light traffic conditions, we use the adaptive coverage control algorithm of section III-C, simplifying the solution of [19], [23].

In the DTRP [14], events appear in the workspace Q according to a space-time Poisson process with rate λ and spatial distribution $\mathbb{P}_z$. When the k^{th} event appears at time t_k, a robot needs to travel to its location Z_k to service it. The robots travel at velocity v according to the kinematic model (1). The time that the k^{th} event spends waiting for a robot to arrive at its location is denoted W_k. The robot then spends a random service time S_k at the event location, where the S_k are iid with finite first and second moments $\bar{s}, \overline{s^2}$. The system time of event k is defined as $\Sigma_k = W_k + S_k, k \geq 1$. The goal is to design policies for the robots that minimize the steady-state system time of the events $\overline{\Sigma} = \limsup_{k \to \infty} E[\Sigma_k]$. Let $\rho = \lambda \bar{s}/n$ denote load factor, i.e., the average fraction of time a robot spends in on-site service. Policies for the DTRP are usually analyzed in two limiting regimes, namely in light traffic conditions ($\rho \to 0^+$) and heavy traffic conditions ($\rho \to 1^-$).

A. *Light Traffic Regime*

Note that we always have [15]

$$\overline{\Sigma} \geq \min_p \mathcal{E}_n(p) + \bar{s}, \tag{25}$$

where $\mathcal{E}_n(p)$ is defined by (4) for $f(x) = x/v$. This bound is tight in light traffic conditions [14], [16], and achieved by the following policy. Let $p^* = [p_1^*, \dots, p_n^*] \in \mathsf{Q}^n$ denote a global minimizer of $\mathcal{E}_n$, called a *multi-median configuration*. In the absence of events, vehicle i waits at the reference position p_i^*. When an event occurs, the agent whose reference position is closest to the event location services it. It then travels back to its reference position p_i^*. As $\rho \to 0^+$, with high probability the agents are at their reference configuration p^* when a new event occurs, and this policy achieves the bound (25).

Assume that $\mathcal{E}_n$ has isolated critical points. The adaptive coverage control policy of Section III-C can then be used to find one of the corresponding robot configurations. In other words, in the absence of event, each robot waits at its reference position $p_{i,k}$. When the k^{th} event occurs at Z_k, the robot whose current reference position is closest to Z_k, say robot j, services the event, and then updates its reference position to $p_{j,k} = \Pi_{\mathsf{Q}}\left[p_{j,k-1} + \gamma_k \frac{1}{v} \frac{Z_k - p_{j,k-1}}{\|Z_k - p_{j,k-1}\|}\right]$. It then travels back toward $p_{j,k}$. Reasoning as in [14], [16], [23], in light traffic the agents are at their reference positions with high probability when an event occurs, and the resulting policy achieves a steady-state system time of $\mathcal{E}_n(\hat{p}) + \bar{s}$, where $\hat{p}$ is a critical point of $\mathcal{E}_n$ to which the stochastic gradient algorithm (15) converges under the assumptions of Theorem 1. Hence we obtain an adaptive policy, which does not achieve the globally minimum system time in general however, unless $n = 1$ since $\mathcal{E}_1$ is convex. The same local optimization is performed adaptively by the light traffic policy described in [23], but an advantage of the stochastic gradient algorithm is that the update rule for the reference positions is simpler to compute. Note that this policy does turns out to be unstable as the load factor ρ increases even if other policies can stabilize the system [16].

B. A Stabilizing Adaptive Policy

Policies adequate for the heavy-traffic regime but requiring $\mathbb{P}_z$ to be known are described in e.g. [16], [19], [45], [46]. The following non-adaptive policy [19], [45], [46], although not the best available, stabilizes the system in heavy-traffic (i.e., as $\rho \to 1^-$). We divide the workspace Q into n regions $\{R_i\}_{i \in [n]}$ such that $\mathbb{P}_z(R_i) = 1/n, i \in [n]$. Robot i only services the events occurring in region i. It does so by forming successive traveling salesman tours (TSP tours) through the event locations falling in his region, and servicing the events in the order of the tours. Recall that a TSP tour through a set of points is the shortest (here, for the Euclidean distance) closed tour through this set of points. While servicing the events in a given tour, new events can occur in region R_i and are backlogged by the robot. Once a tour is finished, the robot forms a new tour through the backlogged events and starts servicing them. When a robot does not have any outstanding event to service, it moves toward the median of its region R_i and stays there as long as no new event occurs in R_i. Note that unless $\{R_i\}_{i \in [n]}$ is a Voronoi partition, which is not compatible in general with the equiprobability property, the resulting configuration in light traffic is not a multi-median configuration, except in the case $n = 1$, and does not offer any performance in the multi-robot case in light traffic. Assuming that $\mathbb{P}_z$ has a density ϕ_z, it is known that under this policy achieves we have the following bounds on the system time in heavy-traffic [19, theorems 4.2, 6.4]

$$\frac{C^*}{n^2} \leq \lim_{\rho \to 1^-} (1-\rho)^2 \overline{\Sigma} \leq \frac{2C^*}{n}, \tag{26}$$

$$\text{where } C^* = C \frac{\lambda \Big(\int_{\mathsf{Q}} \phi_z(z)^{1/2} dz \Big)^2}{v^2} \text{ and } C \approx 0.253.$$

The factor C^*/n^2 is in fact a lower bound on the performance achievable by any policy satisfying a certain fairness condition (called unbiased policies [16]), namely that the steady-state waiting time of an event be independent of its location in the workspace. The policy described above is unbiased. In addition, the right-hand side of (26) can be changed to $2C^*/n^2$ if $\mathbb{P}_z$ is the uniform distribution.

The following adaptive version of this policy stabilizes the system if $\rho < 1$. It does not require the knowledge of any event process parameter such as λ or $\mathbb{P}_z$. To robot i, we associate a fixed point $g_i \in \mathsf{Q}$ and a weight $w_i \in \mathbb{R}$ as in Section IV, a reference position p_i as in Section III-C, and a set of outstanding events to service denoted D_i. We initialize w_i to 0, g_i and p_i to some arbitrary points in Q, and D_i to $\emptyset$. The point g_i remains fixed. The other quantities are updated only at the times where a new event occurs, as follows. When the k^{th} event appears at location Z_k, then

- The robots run the `FloodMin` algorithm with $\hat{d}_j = \|Z_k - g_j\|^2 - w_j, j \in [n]$.
- If robot i has $d_i = \hat{d}_i$, it updates its weight to $w_i \leftarrow w_i - \gamma_{k-1}(n-1)/n$ and its reference position to $p_i \leftarrow \Pi_{\mathsf{Q}} \left[p_i + \gamma_{k-1} \frac{Z_k - p_i}{\|Z_k - p_i\|} \right]$. It then adds Z_k to its set D_i.
- The other robots $j \neq i$ update their weight as $w_j \leftarrow w_j + \gamma_{k-1}/n$ and leave p_j, D_j unchanged.

Each robot $i \in [n]$ then operates according to the following policy

1) As long as $D_i = \emptyset$, travel toward p_i and stay there if p_i is reached.
2) If D_i becomes nonempty
   a) Compute a TSP tour through the points of D_i and set D_i back to $\emptyset$. Start servicing the events in the order of the tour.
   b) Upon completion of a tour, if $D_i \neq \emptyset$, then return to step 2a. If $D_i = \emptyset$, return to step 1.

Theorem 4. *The previously described adaptive policy achieves a steady-state system time sat-*

isfying the heavy traffic performance bound (26), hence stabilizes the system as long as $\rho < 1$*. Moreover if* $n = 1$*, this adaptive policy is optimal in the light traffic regime.*

Proof: As $\rho \to 1$, with high probability the region of each robot is never empty and hence the robot never enters the mode where it goes toward its reference position p_i. By Theorem 3, the partitions $\{R_{i,k}\}_{i\in[n]}$ converge to a power diagram $\{R_i\}_{i\in[n]}$ such that $\mathbb{P}_z(R_i) = 1/n$. Hence the adaptive policy behaves in steady-state as the non-adaptive policy and satisfies (26). In the light traffic regime, in steady state each agent is at the median of its region R_i with high probability when a new event occurs. In particular if $n = 1$ the policy achieves the performance bound (25). ∎

VI. Conclusions

We have discussed robot deployment algorithms for coverage control, spatial partitioning and dynamic vehicle routing problems in the situation where the event distribution is a priori unknown. By adopting the unifying point of view of stochastic gradient algorithms we can derive simple algorithms in each case that locally optimize the objective function (globally in the case of the partitioning algorithms). The coverage control and space partitioning algorithms are combined to provide a fully adaptive solution to the DTRP, with performance guarantees in heavy and light traffic conditions.

Among the issues associated with stochastic gradient algorithms, we point out that they can be slower than their deterministic counterparts and that their practical performance is sensitive to the tuning of the stepsizes γ_k. Many guidelines are available in the literature on stochastic approximation algorithms for the selection of good stepsizes and possibly iterate averaging, see e.g. [41], [42]. In addition, if some prior knowledge about the event distribution is available, it can be leveraged in a straightforward hybrid solution that first deploys the robots using a deterministic gradient algorithm as in the previous work described in the introduction. Once the robots have converged, the adaptive algorithm is used to correct for the modeling errors and environmental uncertainty, exploiting actual observations. Note that the stochastic gradient algorithms can also be used if the distribution $\mathbb{P}_z$ is known, essentially by evaluating integrals such as (6) by Monte-Carlo simulations [38], but this method is only advantageous for q sufficiently large.

Our future work will continue to explore various applications of stochastic approximations to adaptive multi-robot systems, and focus on the experimental evaluation of these algorithms on physical mobile platforms.

Appendix A
Convergence of the Coverage Control Algorithm

In this appendix we collect a number of useful properties of the gradient system

$$\dot{p} = -\nabla\mathcal{E}_n(p), \;\; p(0) \in \mathsf{Q}^n \setminus \mathsf{D}_n, \tag{27}$$

where the distortion function $\mathcal{E}_n$ is defined in (4). As discussed below, this ODE is well defined on $\mathsf{Q}^n \setminus \mathsf{D}_n$. We also consider its extension to Q^n in the form of the differential inclusion

$$\dot{p} \in F(p), \;\; p(0) \in \mathsf{Q}^n, \tag{28}$$

where the the set-valued map F is defined in (17). Following the ODE method [40], we can characterize the asymptotic behavior of the algorithms (14) and (15) as in theorem 1 by studying the properties of these continuous-time dynamical systems. We assume as in section III-C that $f : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is increasing and continuously differentiable. We refer the reader to [8], [9], [25], [38] for previous work on the gradient system (27). In particular, [38] discusses some convergence results for algorithm (14). As pointed out in that paper, the non-differentiability of $\mathcal{E}_n$ creates technical difficulties in the convergence proofs. We handle these difficulties by initially considering the differential inclusion (28) instead of the ODE (27). When the results presented below follow from arguments that can be found in previous work, we simply provide the reference and refer to the detailed proofs in our technical report [39].

Remark 5. Note that even for the ODE (27), we only prove continuity of the right-hand side on $\mathsf{Q}^n \setminus \mathsf{D}_n$. Hence, both for this ODE and the differential inclusion (28), we interpret a solution in the sense of Caratheodory, i.e., an absolutely continuous function $p(t)$ satisfying

$$p(t) = p_0 + \int_0^t y(s)\, ds, \text{ for all } t \in \mathbb{R}, \text{with } y(s) \in F(p(s)) \text{ for all } s.$$

A. Differentiability Properties of $\mathcal{E}_n$

Recall that proposition 1 states that $\mathcal{E}_n$ is continuously differentiable on $\mathbb{R}^n \setminus \mathsf{D}_n$. In general however, $\nabla\mathcal{E}_n$ is discontinuous on the set D_n, see Fig. 4. To discuss more precisely the behavior

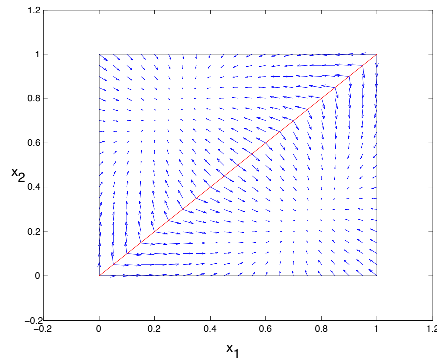

Fig. 4. Vector field for the gradient system (27), with two agents evolving on $[0,1]$ and $\mathbb{P}_z$ uniform on $[0,1]$. The discontinuity on the line $x_1 = x_2$ occurs when the two agents switch side, from $x_1 < x_2$ to $x_1 > x_2$. Note that the vector field is symmetric with respect to this line. The equilibrium occurs at a unique *geometric* point on the line, namely $(1/4, 3/4)$, corresponding to two stationary points for the flow, one for each ordering of the generators.

of the gradient of $\mathcal{E}_n$ as we approach the set D_n, define

$$N(x) = \begin{cases} \|\nabla \mathcal{E}_n(x)\| & \text{if } x \in \mathsf{Q}^n \setminus \mathsf{D}_n \\ \liminf_{y \in W^n \setminus \mathsf{D}_n, y \to x} \|\nabla \mathcal{E}_n(y)\| & \text{if } x \in \mathsf{D}_n. \end{cases}$$

Note that because $\nabla \mathcal{E}_n$ is continuous on $Q^n \setminus \mathsf{D}_n$, the two definitions of N coincide on this set. The proof of the next proposition follows that of [38, lemma 30].

Proposition 2. *Assume that hyperplanes have $\mathbb{P}_z$-measure zero and that $\mathbb{P}_z$ dominates the Lebesgue measure. Then we have $N(x) > 0$ for all $x \in \mathsf{D}_n$. Hence there exists $\delta_0 > 0$ such that*

$$\inf_{x \in B(\mathsf{D}_n, \delta_0) \setminus \mathsf{D}_n} \|\nabla \mathcal{E}_n(x)\| =: \kappa > 0.$$

B. Trajectories of the Gradient System

We now turn to the study of the trajectories of the ODE (27) and the differential inclusion (28). The following general result follows from [38, lemma 33], see also [39].

Proposition 3. *If $x_0 \in \mathsf{Q}^n \setminus \mathsf{D}_n$, a trajectory $t \to x(t)$ of the ODE (27) with $x(0) = x_0$ remains in $\mathsf{Q}^n \setminus \mathsf{D}_n$, i.e., for all $t < \infty$, $x(t) \in \mathsf{Q}^n \setminus \mathsf{D}_n$. Moreover, it converges to a compact connected subset of $\{x \in \mathsf{Q}^n \setminus \mathsf{D}_n : \nabla \mathcal{E}_n = 0\}$.*

We can now show that the trajectories of the ODE never stay in $B(\mathsf{D}_n, \delta_0)$ for a long time.

Corollary 1. *Assume that hyperplanes in $\mathbb{R}^d$ have measure zero, and that $\mathbb{P}_z$ dominates the Lebesgue measure on Q^n. Let $\delta_0 > 0, \kappa > 0$ be defined as in proposition 2, $x_0 \in B(\mathsf{D}_n, \delta_0)$, and let $T = \frac{\max_{x \in \mathsf{Q}^n \cap B(\mathsf{D}_n, \delta_0)} \mathcal{E}_n(x)}{\kappa^2}$. Then a trajectory of the ODE passing through x_0 at time t_1 must exit $B(\mathsf{D}_n, \delta_0)$ at some time $t_2 \leq t_1 + T$.*

Proof: We have, for $t \geq t_1$ and as long as the trajectory remains in $B(\mathsf{D}_n, \delta_0) \setminus \mathsf{D}_n$

$$0 \leq \mathcal{E}_n(x(t)) = \mathcal{E}_n(x_0) - \int_{t_1}^{t} \|\nabla \mathcal{E}_n(x(s))\|^2 ds \leq \max_{x \in \bar{B}(D,\delta)} E(x) - \kappa^2 (t - t_1).$$

Hence the trajectory must exit $B(\mathsf{D}_n, \delta) \setminus \mathsf{D}_n$ at or before the time t_2 given in the theorem. But we know by proposition (3) that it cannot hit D_n at $t_2 < \infty$. Hence it must in fact exit $B(\mathsf{D}_n, \delta)$.

■

The set $\mathcal{H}_n^{\text{ode}}$ defined in (16) is the set of limit points of the ODE (27) by proposition 3. From the definition of F, the set $\mathcal{L}$ of limit points of the differential inclusion (28) consists of the set of limit points of the ODE (27) together with the limit points of the sliding trajectories that start and remain on D_n (since a trajectory leaving D_n does not converge to D_n by proposition (3)). Hence $\mathcal{L} \subset \mathcal{H}_n^{\text{ode}} \cup \mathsf{D}_n$. Moreover, we know by proposition (2) that $\mathcal{H}_n^{\text{ode}} \subset \mathsf{Q}^n \setminus B(\mathsf{D}_n, \delta_0)$ if $\mathbb{P}_z$ dominates the Lebesgue measure.

C. Convergence of the Adaptive Coverage Control Algorithms

We now prove the main convergence theorem 1 for adaptive coverage control.

Proof of theorem 1: We focus on the iterates (14) first. The fact that with probability one, a sequence converges to an compact connected invariant set of the differential inclusion (28) is standard, see e.g. [47, chapter 5]. Consider a sample ω such that $\{p_k(\omega)\}$ converges to such a set, denoted S. Suppose that S is not entirely contained in D_n, and take $a \in S \setminus \mathsf{D}_n$. Then a trajectory of the differential inclusion passing through a at $t = 0$ is in fact a trajectory of the ODE (27), by proposition 3. Because S is invariant, we must then have $\dot{\mathcal{E}}_n(a) := -\|\nabla \mathcal{E}_n(a)\|^2 = 0$, i.e., $a \in \mathcal{H}_n^{\text{ode}}$. This proves the first part of the theorem.

If $\mathbb{P}_z$ dominates the Lebesgue measure, then we know that $\mathcal{H}_n^{\text{ode}}$ and D_n are disconnected by Proposition 2, so S is contained in one of these sets. Choose the sample ω above in the set of probability 1 where the sequence $\{p_k\}_{k \geq 0}$ never hits D_n, and recall the definitions of δ_0 and T from corollary 1. Suppose now that $S \subset \mathsf{D}_n$. Then there exists k_0 such that for all $k \geq k_0$, $p_k \in B(\mathsf{D}_n, \delta_0/4)$. For any $k \geq 0$, denote by $x^k(\cdot)$ the solution of the ODE

(27) starting at p_k (i.e., $x^k(0) = p_k$). Also, denote by $\bar{p}$ the piecewise linear interpolation of the sequence p_k with stepsizes γ_k. Then by [47, chapter 2, lemma 1], there exists $k_1 \geq k_0$ such that for all $k \geq k_1$, we have $\sup_{t \in [t_k, t_k+T]} \|\bar{p}(t) - x^k(t)\| \leq \delta_0/4$, where $t_k := \sum_{l=0}^{k-1} \gamma_l$. In particular, $\|\bar{p}(t_k + T) - x^k(t_k + T)\| \leq \delta_0/4$. Now remark that by Corollary 1, we have $d(x^k(t_k + T), \mathsf{D}_n) > \delta_0$. By possibly increasing k_1, we can assume that there is an iterate $p_{\tilde{k}}$ with $\tilde{k} \geq k$ such that $\|p_{\tilde{k}} - \bar{p}(t_k + T)\| \leq \delta_0/4$. So we have $\|p_{\tilde{k}} - x^k(t_k + T)\| \leq \delta_0/2$, hence $d(p_{\tilde{k}}, \mathsf{D}_n) > \delta_0/2$. But this contradicts our assumptions that $p_{\tilde{k}} \in B(\mathsf{D}_n, \delta_0/4)$. Hence we cannot have $S \subset \mathsf{D}_n$ and so $S \subset \mathcal{H}_n^{\text{ode}}$. This finishes the proof of the theorem for the algorithm (14).

For the projected version (15) of the algorithm, the proof above remains in fact valid. The analysis can indeed be carried in terms of a corresponding projected ODE or differential inclusion, see [41], [47, chapter 5]. But note from proposition 3 that the trajectories of the unprojected ODE never leave Q^n. Hence the projection has no influence on the continuous-time dynamics and the convergence properties remain the same as for the unprojected case. Moreover, the saturation function does not change the convergence properties [41]. ■

APPENDIX B
SPACE PARTITIONING AND OPTIMAL TRANSPORTATION

In this section we prove theorem 2, which forms the basis for the stochastic gradient Algorithm 1 partitioning the workspace between the agents. Compared to the results presented in the recent papers [9], [20], this theorem places weaker assumptions on the cost function $c(x, y)$ and on the target distribution $\mathbb{P}_z$. The main tool on which theorem 2 relies is Kantorovich duality [32]. See also [33], [48], [49] for related results.

proof of theorem 2: We start by relaxing the optimization (19), (20) to the following Monge-Kantorovich optimal transportation problem. Let $P_2 = \sum_{i=1}^n a_i \delta_{g_i}$, so that (20) can be rewritten $\mathbb{P}_z \circ T^{-1} = P_2$. We consider the minimization problem

$$\min_{\pi \in \mathcal{M}(\mathbb{P}_z, P_2)} \int_{\mathsf{Q} \times \mathsf{Q}} c(z, g) d\pi(z, g),$$

where $\mathcal{M}(\mathbb{P}_z, P_2)$ is the set of measures on $\mathsf{Q} \times \mathsf{Q}$ with marginals $\mathbb{P}_z$ and P_2, i.e.,

$$\pi(A \times \mathsf{Q}) = \mathbb{P}_z(A), \quad \pi(\mathsf{Q} \times B) = P_2(B),$$

for all Borel subsets of A, B of Q. In other words, we are considering the problem of transferring some mass from locations distributed according to $\mathbb{P}_z$ to locations distributed according to P_2, and there is a cost $c(x, y)$ for moving a unit of mass from x to y. Then π is a transportation plan from the initial to the final locations, assuming that we allow a unit of mass to be split. The case where this splitting is not allowed, i.e., where we restrict π to be of the form

$$d\pi(z, g) = d\mathbb{P}_z(z)\delta_{T(z)}(g),$$

for some measurable function T, was initially considered by Monge [50], and is exactly our problem (19), (20). In general, the Monge Problem (MP) is more difficult to solve that the Monge-Kantorovich Problem (MKP), but in our case where the target distribution P_2 is discrete, [51, Theorem 3] shows that solving the MKP gives a solution in the form of a transference *function* T, i.e., a solution to the MP, under the assumption A2 of the theorem.

Next, by Kantorovich duality [32], we have

$$\min_{\pi \in \mathcal{M}(\mathbb{P}_z, P_2)} \int_{\mathsf{Q}\times\mathsf{Q}} c(z, g) d\pi(z, g) = \sup_{(\phi, w)\in\Phi_c} \left\{ \int_{\mathsf{Q}} \phi(z)\; d\mathbb{P}_z(z) + \sum_{i=1}^{n} a_i w_i \right\}, \tag{29}$$

where Φ_c is the set of pairs (ϕ, w) with $\phi : \mathsf{Q} \to \mathbb{R}$ in $L^1(\mathsf{Q}, \mathbb{P}_z)$, $w \in \mathbb{R}^n$, such that

$$\phi(z) + w_i \leq c(z, g_i), \tag{30}$$

for $\mathbb{P}_z$-almost all z in Q and for all i in $[n]$. Now for any $w \in \mathbb{R}$, define the function $w^c : \mathsf{Q} \to \mathbb{R}$ such that

$$w^c(z) = \min_{i\in[n]} \{c(z, g_i) - w_i\}.$$

From the definition of Φ_c, we can then without loss of generality restrict the supremum on the right-hand side of (29) to pairs of the form (w^c, w). Combining this with the previous remark on the Monge solution to the Monge-Kantorovich problem, we get

$$\min_{T:\mathsf{Q}\to\{g_1,\ldots,g_n\}} \int_{\mathsf{Q}} c(z, T(z))\mathbb{P}(dz) = \sup_{w\in\mathbb{R}^n} \left\{ \int_{\mathsf{Q}} \min_{i\in[n]} \{c(z, g_i) - w_i\}\; \mathbb{P}_z(dz) + \sum_{i=1}^{n} a_i w_i \right\}. \tag{31}$$

Hence the value of the optimization problem is equal to the supremum of the function h defined in (21). The fact that the supremum is attained in the right hand side of (31) follows from e.g. [32, Theorem 2.3.12] under our majorization assumption A1 for c.

define $\tilde{c}(z) = \min_{i\in[n]}\{c(z, g_i)\}$, and to note that $\tilde{c}(z)$ is bounded on Q compact.

It is easy to see that h is concave since $w \to \min_{i\in[n]}\{c(z, g_i) - w_i\}$ is concave for all z as the minimum of affine functions, and the integration with respect to z preserves concavity. Finally, for $w^1, w^2 \in \mathbb{R}^n$, we have

$$h(w^2)-h(w^1) = \int_{\mathsf{Q}} \min_{i\in[n]}\{c(z,g_i)-w_i^2\}\,\mathbb{P}_z(dz) - \int_{\mathsf{Q}} \min_{i\in[n]}\{c(z,g_i)-w_i^1\}\,\mathbb{P}_z(dz) + \sum_{i=1}^n a_i(w_i^2-w_i^1).$$

Denoting T^1 an assignment that is optimal for w^1, we have then, for all $z \in \mathsf{Q}$,

$$\min_{i\in[n]}\{c(z,g_i)-w_i^2\} \le c(z,T^1(z)) - w_i^2,$$

and so

$$h(w^2)-h(w^1) \le -\sum_{i=1}^n \mathbb{P}_z(\tilde{V}_i^c(w^1))(w_i^2-w_i^1) + \sum_{i=1}^n a_i(w_i^2-w_i^1).$$

But this inequality exactly says that $[a_1 - \mathbb{P}_z(\tilde{V}_1^c(w^1)), \dots, a_n - \mathbb{P}_z(\tilde{V}_n^c(w^1))]^T$ is a supergradient of h at w^1. ■

REFERENCES

[1] A. Howard, M. Mataric, and G. Sukhatme, "Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem," in *Proceedings of the 6th International Symposium on Distributed Autonomous Robotics Systems (DARS02)*, Fukuoka, Japan, 2002.

[2] P. Ögren, E. Fiorelli, and N. Leonard, "Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment," *IEEE Transactions on Automatic Control*, vol. 49, no. 8, pp. 1292–1302, August 2004.

[3] F. Bullo, J. Cortés, and S. Martinez, *Distributed Control of Robotic Networks*. Princeton University Press, 2009.

[4] J. Cortés, S. Martínez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, April 2004.

[5] R. Gray, "Vector quantization," *IEEE ASSP Magazine*, vol. 1, no. 2, pp. 4–29, April 1984.

[6] Z. Drezner, Ed., *Facility Location: A Survey of Applications and Methods*, ser. Springer Series in Operations Research. New York: Springer Verlag, 1995.

[7] S. Lloyd, "Least-squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, March 1982.

[8] Q. Du, V. Faber, and M. Gunzburger, "Centroidal Voronoi tessellations: Applications and algorithms," *SIAM Review*, vol. 41, no. 4, pp. 637–676, December 1999.

[9] J. Cortés, S. Martínez, and F. Bullo, "Spatially-distributed coverage optimization and control with limited-range interactions," *ESAIM: Control, Optimisation and Calculus of Variations*, vol. 11, no. 4, pp. 691–719, 2005.

[10] W. Li and C. G. Cassandras, "Distributed cooperative control of sensor networks," in *Proceedings of the Conference on Decision and Control*, Seville, Spain, December 2005, pp. 2542–2547.

[11] K. Guruprasad and D. Ghose, "Generalized Voronoi partition based multi-agent search using heterogeneous sensors," August 2009, http://arxiv.org/abs/0908.2683v1.

[12] L. Pimenta, V. Kumar, R. Mesquita, and G. Pereira, "Sensing and coverage for a network of heterogeneous robots," in *Proceedings of the Conference on Decision and Control*, 2008.

[13] M. Schwager, D. Rus, and J.-J. Slotine, "Decentralized, adaptive coverage control for networked robots," *The International Journal of Robotics Research*, vol. 28, no. 3, pp. 357–375, March 2009.

[14] D. Bertsimas and G. J. Van Ryzin, "A stochastic and dynamic vehicle routing problem in the Euclidean plane," *Operations Research*, vol. 39, no. 4, pp. 601–615, July-August 1991.

[15] ——, "Stochastic and dynamic vehicle routing in the Euclidean plane with multiple capacited vehicles," *Operations Research*, vol. 41, no. 1, pp. 60–76, 1993.

[16] D. J. Bertsimas and G. J. Van Ryzin, "Stochastic and dynamic vehicle routing with general demand and interarrival time distributions," *Advances in Applied Probability*, vol. 25, no. 4, pp. 947–978, December 1993.

[17] E. Frazzoli and F. Bullo, "Decentralized algorithms for vehicle routing in a stochastic time-varying environment," in *Proceedings of the Conference on Decision and Control*, 2004.

[18] K. Savla, "Multi-UAV systems with motion and communication constraints," Ph.D. dissertation, University of California, Santa Barbara, 2007.

[19] M. Pavone, E. Frazzoli, and F. Bullo, "Adaptive and distributed algorithms for vehicle routing in a stochastic and dynamic environment," 2009, http://www.mit.edu/∼pavone/papers/Pavone.Frazzoli.ea.TAC09.pdf, submitted for publication.

[20] M. Pavone, A. Arsie, E. Frazzoli, and F. Bullo, "Equitable partitioning policies for mobile robotic networks," Massachusetts Institute of Technology, Tech. Rep., 2009, submitted for publication.

[21] J. Cortés, "Coverage control and spatial load balancing by robotic sensor networks," *IEEE Transactions on Automatic Control*, vol. 55, no. 3, pp. 749–754, March 2010.

[22] J. Choi and R. Horowitz, "Learning coverage control of mobile sensing agents in one-dimensional stochastic environments," *IEEE Transactions on Automatic Control*, 2009, conditionally Accepted.

[23] A. Arsie, K. Savla, and E. Frazzoli, "Efficient routing algorithms for multiple vehicles with no explicit communications," *IEEE Transactions on Automatic Control*, vol. 54, no. 10, pp. 2302–2317, October 2009.

[24] A. Benveniste, M. Metivier, and P. Priouret, *Adaptive Algorithms and Stochastic Approximations*. Springer, 1990.

[25] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. Wiley, 2000.

[26] R. M. Gray and D. L. Neuhoff, "Quantization," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2325–2383, October 1998.

[27] J. Mac Queen, "Some methods for the classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, L. Le Cam and J. Neyman, Eds. University of California Press, 1967, pp. 281–297.

[28] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybernetics*, vol. 43, pp. 59–69, 1982.

[29] F. Aurenhammer, F. Hoffmann, and B. Aronov, "Minkowski-type theorems and least-squares clustering," *Algorithmica*, vol. 20, pp. 61–76, 1998.

[30] B. Aronov, P. Carmi, and M. J. Katz, "Minimum-cost load-balancing partitions," in *Proceedings of the 22nd ACM Symposium on Computational Geometry*, Sedona, AZ, 2006, pp. 301–308.

[31] J. Le Ny and G. J. Pappas, "Adaptive robot deployment algorithms," University of Pennsylvania, Tech. Rep., March 2010 2010, http://repository.upenn.edu/ese_reports/4/.

[32] S. T. Rachev and L. Rüschendorf, *Mass Transportation Problems*. Springer Verlag, 1998, vol. 1: Theory.

[33] W. Gangbo and R. McCann, "The geometry of optimal transportation," *Acta Mathematica*, vol. 177, no. 2, pp. 113–161, 1996.

[34] L. Rüschendorf and L. Uckelmann, "On optimal multivariate couplings," in *Distributions with Given Marginals and Moment Problems*, V. Beneš and J. Štěpán, Eds. Kluwer Academic Publishers, 1997, pp. 261–273.

[35] F. Aurenhammer, "Voronoi diagrams - a survey of a fundamental geometric data structure," *CM Computing Surveys*, vol. 23, no. 3, pp. 345–405, September 1991.

[36] N. A. Lynch, *Distributed algorithms*. Morgan Kaufmann, 1996.

[37] S. Fekete, J. Mitchell, and K. Beurer, "On the continuous Fermat-Weber problem," *Operations Research*, vol. 53, no. 1, pp. 61–76, January-February 2005.

[38] G. Pagès, "A space quantization method for numerical integration," *Journal of Computational and Applied Mathematics*, vol. 89, no. 1, pp. 1–38, March 1997.

[39] J. Le Ny and G. J. Pappas, "Adaptive algorithms for coverage control and space partitioning in mobile robotic networks," University of Pennsylvania, Tech. Rep., 2010.

[40] L. Ljung, "Analysis of recursive stochastic algorithms," *IEEE Transactions on Automatic Control*, vol. 22, no. 4, pp. 551–575, Augus 1977.

[41] H. J. Kushner and G. G. Yin, *Stochastic Approximation and Recursive Algorithms and Applications*, 2nd ed. Springer, 2003.

[42] J. C. Spall, *Introduction to Stochastic Search and Optimization*. Wiley, 2003.

[43] J. Cortés, "Discontinuous dynamical systems - a tutorial on solutions, nonsmooth analysis, and stability," *IEEE Control Systems Magazine*, vol. 28, no. 3, pp. 36–73, 2008.

[44] "CGAL, Computational Geometry Algorithms Library," http://www.cgal.org.

[45] H. Xu, "Optimal policies for stochastic and dynamics vehicle routing problems," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, 1995.

[46] J. D. Papastavrou, "A stochastic and dynamic routing policy using branching processes with state dependent immigration," *European Journal of Operations Research*, vol. 95, pp. 167–177, 1996.

[47] V. Borkar, *Stochastic Approximation: A Dynamical Systems Viewpoint*. Cambridge University Press, 2008.

[48] T. Abdellaoui, "Distances de deux lois dans les espaces de Banach," Ph.D. dissertation, Université de Rouen, 1993.

[49] L. Rüschendorf and L. Uckelmann, "Numerical and analytical results for the transportation problem of Monge-Kantorovich," *Metrika*, vol. 51, pp. 245–258, 2000.

[50] G. Monge, "Mémoire sur la théorie des déblais et des remblais."

[51] J. A. Cuesta-Albertos and A. Tuero-Díaz, "A characterization for the solution of the Monge-Kantorovich mass transference problem," *Statistics and Probability Letters*, vol. 16, pp. 147–152, 1993.