

Adaptive Deployment of Mobile Robotic Networks

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

Abstract—This paper considers 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. Moreover, it is assumed that the event location distribution is *a priori* unknown, and can only be progressively inferred from the observation of the actual event occurrences. Three classes of problems are discussed in detail: coverage control problems, spatial partitioning problems, and dynamic vehicle routing problems. In each case, distributed stochastic gradient algorithms optimizing the performance objective are presented. The stochastic gradient view simplifies and generalizes previously proposed solutions, and is applicable to new complex scenarios, such as adaptive coverage involving heterogeneous agents. Remarkably, these algorithms often take the form of simple distributed rules that could be implemented on resource-limited platforms.

Index Terms—Adaptive algorithms, coverage control problems, dynamic vehicle routing problems, partitioning algorithms, potential field based motion planning, stochastic gradient descent algorithms.

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 missions with 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 basic version of the coverage control problem has inspired many variations, e.g., considering limited communication and sensing radii [8], [9], heterogeneous sensors [10], obstacles and non-point robots [11], or applications to field estimation problems [12]. It is also related to certain vehicle routing problems, notably the Dynamic Traveling Repairman Problem (DTRP) [13]–[15], as discussed by Frazzoli and Bullo in [16] and several subsequent papers [17], [18]. Another related problem is the space partitioning problem [19], [20], 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 distribution of the events, and the optimization algorithms require the knowledge of this distribution [4], [16], [19], [20]. 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 these observations. Indeed, recently some coverage control algorithms [12], [21] and vehicle routing algorithms [18], [22] have been developed to work in the absence of *a priori* knowledge of the event location distribution.

An essential idea of our work is that deployment problems with stochastic uncertainty can 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 location distribution*. We call these algorithms *adaptive*, in analogy with the engineering literature on adaptive systems [23]. 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 limited sensing, computational and communication capabilities.

Specifically, we first discuss in Section III certain stochastic gradient algorithms that adaptively optimize coverage control objectives. These algorithms generalize to new complex multi-

Manuscript received November 04, 2010; revised November 30, 2011; accepted August 16, 2012. Date of publication August 24, 2012; date of current version February 18, 2013. This work was supported by the ONR-MURI HUNT award N00014-08-1-0696. Recommended by Associate Editor M. Egerstedt.

J. Le Ny is with the Department of Electrical Engineering, École Polytechnique de Montréal, QC H3T 1J4, Montreal, Canada (e-mail: jerome.le-ny@polymtl.ca).

G. J. Pappas is with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104 USA (e-mail: pappasg@seas.upenn.edu); Phone: (215) 898-9780; Fax: (215) 573-2068.

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/TAC.2012.2215512

agent deployment problems and we 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 [24], [25]. 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 [26]–[28]. Finally in Section V we present the first fully adaptive algorithm for the DTRP. In light traffic conditions, the policy reduces to the coverage control algorithm of Section III, and is simpler than the previous algorithm presented in [22]. In heavy traffic conditions, it relies on the partitioning algorithm of Section IV. This algorithm complements the recent work of Pavone *et al.* [18], in which the knowledge of the event location distribution is required in the heavy traffic regime.

II. PRELIMINARIES

Notation

We denote $[n] := \{1, \dots, n\}$. Throughout the paper all random elements are defined on a generic probability space (Ω, \mathcal{F}, P) , with the expectation operator corresponding to P denoted E . We abbreviate “independent and identically distributed” by iid, and “almost surely” by a.s. 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 $\delta_{x_0}(S) = \mathbf{1}_S(x_0)$ for all Borel subsets S of X . We denote the distance from a point $x \in X$ to a set S by $d(x, S) := \inf_{y \in S} d(x, y)$, and we set $d(x, \emptyset) = +\infty$. A sequence of points $\{x_k\}_{k \geq 0}$ in X is said to converge to a set $S \subset X$ if $d(x_k, S) \rightarrow 0$ as $k \rightarrow \infty$. 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 $B(x, r)$.

A solution of a differential equation $\dot{x} = h(x)$ or of a differential inclusion $\dot{x} \in \mathcal{H}(x)$ [29] is interpreted in the sense of Caratheodory, i.e., as an absolutely continuous function $x(t)$ satisfying

$$x(t) = x_0 + \int_0^t y(s) ds, \text{ for all } t \in \mathbb{R}, \text{ with } y(s) = h(x(s)) \\ \text{or } y(s) \in \mathcal{H}(x(s)) \text{ for all } s.$$

Finally, a set I is invariant under a differential inclusion $\dot{x} \in \mathcal{H}(x)$ if for all $x_0 \in I$, there exist *some* solution $x(t), t \in (-\infty, \infty)$, with $x(0) = x_0$, that lies entirely in I .

A. Robot Network Model

We consider a group of n robots evolving in \mathbb{R}^q , for some $q \geq 1$. We denote the robot positions at time $t \in \mathbb{R}_{\geq 0}$ by $p(t) =$

$[p_1(t), \dots, p_n(t)] \in (\mathbb{R}^q)^n$. For simplicity, we assume that the robots follow a simple kinematic model

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

where v_i is a positive constant and 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., (17). In addition, the robots are assumed to perform computations and to communicate instantaneously. Finally, we define

$$D_n = \left\{ p = [p_1, \dots, p_n] \in (\mathbb{R}^q)^n \mid \begin{aligned} & p_i = p_j \text{ for some } 1 \leq i < j \leq n \end{aligned} \right\}. \quad (2)$$

Hence D_n is the set of configurations where at least two robots occupy the same position.

B. Geometric Optimization

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

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

That is, V_i is the set of points 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 [30]. Two points p_i and p_j or their indices i, j (with $i \neq j$) are called Voronoi neighbors if the boundaries of their Voronoi cells intersect, i.e., if $V_i(p) \cap V_j(p) \neq \emptyset$.

For a function $c : \mathbb{R}^q \times \mathbb{R}^q \rightarrow \mathbb{R}$, a vector $w = [w_1, \dots, w_n] \in \mathbb{R}^n$, and $p = [p_1, \dots, p_n] \in (\mathbb{R}^q)^n \setminus D_n$, define for all $i \in [n]$ the generalized Voronoi cell of the pair (p_i, w_i) with respect to c by

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

We also write $V_i^c(\mathcal{G}, w) := V_i^c(p, w)$ for the set $\mathcal{G} = \{p_1, \dots, p_n\}$. The point p_i is called the generator and w_i the weight of the cell $V_i^c(p, w)$, and $\{V_i^c\}_{i \in [n]}$ a generalized Voronoi diagram. Intuitively $c(z, p)$ represents a distance or cost between the points z and p , and in practice takes the form $c(z, p) = f(\|z - p\|)$, with f an increasing function. In particular for $f(x) = x^2$, the generalized Voronoi diagram is called a power diagram [30], [31], and the generalized Voronoi cell a power cell. Like Voronoi cells, power cells are polyhedra, although possibly empty [31]. Notice from (3) that for a given configuration p , 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.

C. 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 the set $\arg \min_{i \in [n]} \hat{d}_i$.

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 diam . Alternatively, they know the number n of robots in the system, in which case we take $\text{diam} = n$ below.

In a synchronous network the problem can be solved by the FloodMin algorithm [32, section 4.1.2]. Each robot maintains a record in a variable d_i of the minimum number it has seen so far, with $d_i = \hat{d}_i$ initially. At each round, it sends this minimum to all its neighbors. The algorithm terminates after diam rounds. The agents that still have $d_i = \hat{d}_i$ 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 [32, 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 $\lim_{t \rightarrow \infty} p_i(t)$, $i \in [n]$, minimize the following performance measure capturing the quality of coverage of certain events:

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

where $f: \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ is an increasing continuously differentiable function. The random variable Z represents the location of an event of interest occurring in the workspace. To interpret (4), the cost of servicing an event at location z with a robot at location p_i is measured by $f(\|p_i - z\|)$, and an event must be serviced by the robot closest to the location of this event. For example, in monitoring applications, $f(\|p_i - z\|)$ can measure the degradation of the sensing performance with the distance to the event [4]. In vehicle routing problems, this cost might be the time it takes a robot to travel to the event location, i.e., $f(\|p_i - z\|) = \|p_i - z\|/v_i$, assuming enough time between successive events, see Section V.

For simplicity, we assume in this section that the probability distribution $\mathbb{P}_z := P \circ Z^{-1}$ of Z has support contained in a compact convex set Q with nonempty interior. We also generally make the following assumption.

Assumption 1: Hyperplanes in \mathbb{R}^q have \mathbb{P}_z -measure zero.

Note that Assumption 1 implies that points also have measure zero, and in particular the support of \mathbb{P}_z is infinite. The following result, whose proof can be found in [33, Proposition 9], [34], provides an expression for the derivatives of \mathcal{E}_n , useful for optimization purposes. 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 . We also adopt the convention $0/\|0\| := 0$.

Proposition 1: Under Assumption 1, \mathcal{E}_n is Lipschitz continuous on compact sets and continuously differentiable on $(\mathbb{R}^q)^n \setminus D_n$, with partial derivatives

$$\frac{\partial \mathcal{E}_n}{\partial p_i} \Big|_p = \int_{V_i(p)} f'(\|p_i - z\|) \frac{p_i - z}{\|p_i - z\|} \mathbb{P}_z(dz). \quad (5)$$

Now let us suppose, as in [4] and most of the subsequent literature, that the event location distribution \mathbb{P}_z is known. Using (5), one can then implement a gradient descent algorithm to locally minimize the objective (4)[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 $p_k := p(kT) = [p_{1,k}, \dots, p_{n,k}]$. The robots start at $p_0 = [p_{1,0}, \dots, p_{n,0}]$ at $t = 0$ and update their positions according to

$$p_{i,k+1} = p_{i,k} - \gamma_k \frac{\partial \mathcal{E}_n}{\partial p_i} \Big|_{p_k}, \quad (6)$$

where γ_k is an appropriately chosen sequence of decreasing or small constant positive stepsizes. We ignore for the moment the issue of non-differentiability on D_n as well as the minor modifications required to accommodate velocity constraints in (6). The agents can implement (6) to 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 [35]. Nevertheless, an interesting property of the gradient descent algorithm (6) for the coverage control problem is that it can be implemented in a distributed manner by the robots, by exploiting the form of the expression (5). In particular, each agent can update its position at each period according to (6) 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 (5). Even in a large network, a single robot has typically only few Voronoi neighbors [31], which allows for a scalable and distributed implementation of the gradient descent algorithm.

Remark 1: The specific case where $f(x) = x^2$ is considered in [4] in more details. In this case (5) gives

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

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

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

Then control law (6), i.e.,

$$\begin{aligned} p_{i,k+1} &= p_{i,k} - \gamma_k \frac{\partial \mathcal{E}_n}{\partial p_i} \Big|_{p_k} \\ &= p_{i,k} - 2\gamma_k \mathbb{P}_z(V_i(p_k))(p_{i,k} - C_{V_i(p_k)}), \end{aligned} \quad (8)$$

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

Note that the computation of the updates (6) requires \mathbb{P}_z to be perfectly known. The minimization of (4) is then essentially an open-loop optimization problem, and the network can reach its desired configuration *before any event occurs*. However, the algorithm does not provide any mechanism to adapt the configuration based on the actual observations of where the events occur, which is critical in practice as \mathbb{P}_z is rarely available. In the next section, we show how to generally address this issue by using stochastic gradient algorithms. Section III-C applies the method specifically to the adaptive coverage control problem.

B. Stochastic Gradient Algorithms

Suppose that we wish to minimize a function G defined on \mathbb{R}^m for some $m \geq 0$, of the form

$$\begin{aligned} G(x) &= E[g(x, Z)] = \int_{\Omega} g(x, Z(\omega)) P(d\omega) \\ &= \int_Z g(x, z) \mathbb{P}_z(dz), \end{aligned} \quad (9)$$

such as \mathcal{E}_n defined in (4) for example. The space Z in (9) is the range of the random variable Z . In contrast to the previous subsection, we now assume that \mathbb{P}_z is unknown, so that the expectation (9) cannot be computed directly. Suppose that g is differentiable with respect to x , for \mathbb{P}_z -almost all z , and denote its gradient $\nabla_x g(x, z) := \partial g(x, z) / \partial x$. Finally, assume that we can observe random variables Z_k , $k \geq 1$, iid with distribution \mathbb{P}_z . Consider then the stochastic recursive algorithm

$$x_{k+1} = x_k - \gamma_k \nabla_x g(x_k, Z_{k+1}), \quad (10)$$

which can be rewritten in the form

$$x_{k+1} = x_k + \gamma_k (h(x_k) + D_{k+1}), \quad (11)$$

with $h(x) := -E[\nabla_x g(x, Z_1)|x]$ and $D_{k+1} = -\nabla_x g(x_k, Z_{k+1}) + E[\nabla_x g(x_k, Z_{k+1})|x_k]$. Define for $k \geq 0$ the filtration $\mathcal{F}_k := \sigma(x_0, D_i, 1 \leq i \leq k)$, i.e., an increasing family $\mathcal{F}_k \subset \mathcal{F}_l$ for $k \leq l$ of sub- σ -algebras of \mathcal{F} . Then $\{D_k\}_{k \geq 1}$ is a *martingale difference sequence* (MDS) with respect to $\{\mathcal{F}_k\}_{k \geq 0}$, as explained in the following definition.

Definition 1: Let $\{\mathcal{F}_k\}_{k \geq 0}$ be a filtration. A sequence of random variables $\{D_k\}_{k \geq 1}$ is called a martingale difference sequence with respect to $\{\mathcal{F}_k\}_{k \geq 0}$ if D_k is measurable with respect to \mathcal{F}_k , $E[\|D_k\|] < \infty$, and we have $E[D_k | \mathcal{F}_{k-1}] = 0$, for all $k \geq 1$.

Intuitively, the MDS $\{D_k\}_{k \geq 1}$ plays the role of a zero-mean noise. By the ODE method [36], we can expect that asymptotically, under the condition

$$\gamma_k \geq 0, \quad \sum_{k=0}^{\infty} \gamma_k = +\infty, \quad \sum_{k=0}^{\infty} \gamma_k^2 < +\infty, \quad (12)$$

on the stepsizes, which holds for $\gamma_k = 1/(1+k)$ for example, the sequence $\{x_k\}_{k \geq 0}$ in (11) almost surely approaches the trajectories of the ODE

$$\dot{x} = h(x). \quad (13)$$

Now assume that it is valid to exchange expectation and derivation, as follows

$$\begin{aligned} -\nabla G(x) &= -\nabla E[g(x, Z_1)|x] \\ &= -E[\nabla_x g(x, Z_1)|x] = h(x). \end{aligned} \quad (14)$$

Identity (14) can often be proved using the dominated convergence theorem, see, e.g., [37, Theorem 5.1]. Then the ODE (13) describes a gradient flow and so in fact under mild assumptions the trajectories and therefore the sequence $\{x_k\}_{k \geq 0}$ approach the critical points of G . Moreover, we can often expect convergence to the set of local minima of G almost surely

[38, chapter 4]. In conclusion, the algorithm (11) allows us to reach such a minimum even though \mathbb{P}_z is unknown, as long as we have access to realizations of the random variables Z_k .

We now capture the intuition above more formally, including the situation where the function G is not everywhere differentiable, as in Proposition 1. Consider a stochastic algorithm

$$x_{k+1} = x_k + \gamma_k (h_k + D_{k+1}), \quad \forall k \geq 0, \quad (15)$$

where the stepsizes γ_k satisfy (12), $\{D_k\}_{k \geq 1}$ is an MDS with respect to the filtration $\mathcal{F}_k := \{x_l, h_l, D_l, l \leq k\}$, $k \geq 0$, and h_k is specified in the following theorems.

Theorem 1: Assume that G is continuously differentiable on $\mathbb{R}^m \setminus S$, with S a set of Lebesgue measure zero. Introduce the Filippov set-valued map [29]

$$\mathcal{H}(x) = \begin{cases} \{-\nabla G(x)\}, & x \notin S, \\ \bigcap_{\delta > 0} \overline{\text{co}} \left(\bigcup_{\hat{x} \in B(x, \delta) \setminus S} \{-\nabla G(\hat{x})\} \right), & x \in S, \end{cases} \quad (16)$$

where $\overline{\text{co}}$ denotes the closed convex hull. Consider the recurrence (15) with $h_k \in \mathcal{H}(x_k)$, for all $k \geq 0$. Assume that for some positive constants K_1, K_2 we have

$$\begin{aligned} \sup_{h \in \mathcal{H}(x)} \|h\| &\leq K_1(1 + \|x\|), \quad \forall x \in \mathbb{R}^m, \\ E[\|D_{k+1}\|^2 | \mathcal{F}_k] &\leq K(1 + \|x_k\|^2), \quad \text{a.s.}, \quad \forall k \geq 0, \end{aligned}$$

and that $\sup_{k \geq 0} \|x_k\| < \infty$, a.s. Then the sequence $\{x_k\}_{k \geq 0}$ converges almost surely to a connected subset of $\{x \in \mathbb{R}^m \setminus S | \nabla G(x) = 0\} \cup S$, invariant for the differential inclusion $\dot{x} \in \mathcal{H}(x)$.

Theorem 2: Assume that G is convex and admits a minimum on \mathbb{R}^m . Consider the recurrence (15) with h_k a subgradient of G at x_k , for all $k \geq 0$. Assume that there exists a positive constant K such that $E[\|h_k + D_{k+1}\|^2 | \mathcal{F}_k] \leq K$, for all $k \geq 0$. Then the sequence $\{x_k\}_{k \geq 0}$ converges almost surely to some point minimizing G .

The proofs of these theorems are standard and not repeated here, see [38], [39, chapter 5], [40, Proposition 8.2.6. p. 480] and the proof of Theorem 3 in Appendix A-C. Note that in many applications, the stepsizes γ_k are chosen to converge to a small positive constant instead of satisfying (12), which allows tracking of the equilibria of the gradient flow if the problem parameters (e.g., \mathbb{P}_z) change with time. In this case, one typically obtains convergence to a neighborhood of a critical point [23]. 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 [37], [39].

C. Adaptive Coverage Control

We now consider the following modification of the coverage control problem. The events occur randomly in the workspace, with event $k \geq 1$ occurring at time $t_k > 0$ and location $Z_k \in Q$. 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 as a result the deterministic gradient descent algorithm (6) cannot be implemented. We work under Assumption 1, so that the gradient expression (5) holds.

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

$$p_{i,k+1} = p_{i,k} + u_{i,k}, \quad |u_{i,k}| \leq v_{i,k}, \quad \forall k \in \mathbb{Z}_{\geq 0}, \quad \forall i \in [n], \quad (17)$$

where $u_{i,k} \in \mathbb{R}^q$ is a control input for the interval $[t_k, t_{k+1})$. For example, if the robot dynamics follow the model (1) and if 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 $Z_k \in Q$, 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 a new reference position $p_{i,k}$. The following reference position updates implement the stochastic gradient algorithm (10) to minimize the coverage objective (4). First, for a vector $u \in \mathbb{R}^q$ and a scalar $b > 0$, define 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 update rule

$$p_{i,k+1} = \begin{cases} \Pi_Q \left[p_{i,k} + \text{sat} \left[\gamma_k f'(\|p_{i,k} - Z_{k+1}\|) \frac{Z_{k+1} - p_{i,k}}{\|Z_{k+1} - p_{i,k}\|} \right]_{v_{i,k}} \right] & \text{if } i \in \arg \min_{j \in [n]} \|p_{j,k} - Z_{k+1}\|, \\ p_{i,k} & \text{otherwise,} \end{cases} \quad (18)$$

where Π_Q is the orthogonal projection on the convex set Q . Note that the situation where several robots are at equal distance from Z_k and simultaneously update their reference position occurs with probability zero under Assumption 1. To justify (18) based on the discussion in the previous subsection, let $g(p, z) = \min_{i \in [n]} f(\|p_i - z\|)$, i.e., $g(p, z) = f(\|p_{i^*} - z\|)$ for $z \in V_{i^*}(p)$. Then we have

$$\frac{\partial g}{\partial p_i}(p, z) = \begin{cases} f'(\|p_i - z\|) \frac{p_i - z}{\|p_i - z\|}, & \text{if } z \in \text{Int}(V_i(p)) \setminus \{p_i\} \\ 0, & \text{if } z \notin V_i(p). \end{cases}$$

Moreover let us define $(\partial g / \partial p)(p, z)$ arbitrarily for z on the Voronoi cell boundaries and at the points p_i . These sets have \mathbb{P}_z -measure zero under Assumption 1, and hence do not contribute to the integral

$$\begin{aligned} E \left[\frac{\partial g}{\partial p_i}(p, Z) \right] &= \int_{V_i(p)} f'(\|p_i - z\|) \frac{p_i - z}{\|p_i - z\|} \mathbb{P}_z(dz) \\ &= \frac{\partial \mathcal{E}_n}{\partial p_i}(p), \quad \text{for } p \notin D_n. \end{aligned}$$

In other words, Proposition 1 precisely says that the identity (14) is valid for \mathcal{E}_n in $(\mathbb{R}^q)^n \setminus D_n$. Note also that almost surely the update rule (18) never results in two robots landing on the same position as long as $q \geq 2$ and the updated reference position

before projection remains in Q , because this would require Z_{k+1} to fall on the line passing through these two robot reference positions. This can be achieved for $q = 1$ or for a reference position projected on the boundary of Q as well, by a small random perturbation of the sequence γ_k [38, Chapter 2]. Hence we can assume in the following that almost surely $p_k \notin D_n$ for all $k \geq 1$. Moreover, the projection Π_Q and the saturation nonlinearity do not change the convergence properties of the algorithm, see Appendix A. Therefore, (18) is essentially the stochastic gradient descent update rule (10).

It is interesting to compare the implementation complexity of algorithm (18) with that of the corresponding deterministic gradient descent update based on (5), (6). The deterministic, model-based algorithm requires that each agent maintains communication with its Voronoi neighbors and knows their position in order to determine the boundaries of its Voronoi cell and compute the integral (5). Even in the quadratic case (8), this scheme can be difficult to implement. In contrast, no Voronoi cell computation or integration and no detailed knowledge of the position of the neighbors is required by (18), which only needs a distributed mechanism to find which robot is the closest to the target when it appears. This can be done in a distributed way via the FloodMin algorithm described in Paragraph Section 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. Clearly there are other ways to implement the rule (18). For example, we could let all the robots travel to the event location at the same speed, as in [22], 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.

1) Special Cases: If we specialize (18) to the least-squares coverage control problem with $f(x) = x^2$ and ignore the saturation function, 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], [41], [42]. 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 (18) is simply $p_{i,k+1} = p_{i,k} + \gamma_k((Z_{k+1} - p_{i,k}) / \|Z_{k+1} - p_{i,k}\|)$ for the closest robot. It is used in the vehicle routing application discussed in Section V.

Remark 2: For certain distributions and initial robot positions outside of the support set of the distribution \mathbb{P}_z , it is possible that by following (18), some agents never move. The issue also arises with the deterministic algorithm, since the gradient (7) vanishes if $\mathbb{P}_z(V_i(p_k)) = 0$. 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 (18) rather than only the closest agent. The goal of this transient modification is to bring all the robots within the support set of the event distribution. It is either stopped at some finite time or discounted by a stepsize decreasing much faster than γ_k , thereby not impacting the convergence results [38].

We now state a convergence result for the update law (18) to the set of critical points of the objective \mathcal{E}_n , i.e., to

$$C_n = \{p \in Q^n \setminus D_n \mid \nabla \mathcal{E}_n(p) = 0\}. \quad (19)$$

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 (18) can be found in [33] and leaves open the question of non-convergence to D_n . In contrast to that paper, we cope with the non-differentiability on D_n by introducing the Filippov set-valued map \mathcal{H}_n as in (16)

$$\mathcal{H}_n(x) = \begin{cases} \{-\nabla \mathcal{E}_n(x)\}, & x \notin D_n, \\ \bigcap_{\delta > 0} \overline{\text{co}} \left(\bigcup_{\hat{x} \in B(x, \delta) \setminus D_n} \{-\nabla \mathcal{E}_n(\hat{x})\} \right), & x \in D_n. \end{cases} \quad (20)$$

We also need the following definition. A Borel measure μ on \mathbb{R}^q is said to dominate the Lebesgue measure λ if $\lambda(A) = 0$ for all Borel sets A such that $\mu(A) = 0$.

Theorem 3: *Let the stepsizes γ_k satisfy (12), $p_0 \in Q^n \setminus D_n$, and suppose that Assumption 1 holds. Then, by following the algorithm (18), the sequence $\{p_k\}_{k \geq 0}$ of robot positions converges almost surely to a compact connected subset of $C_n \cup (D_n \cap Q^n)$, invariant for the differential inclusion $\dot{p} \in \mathcal{H}_n(p)$.*

If in addition \mathbb{P}_z dominates the Lebesgue measure on Q , then the sequence $\{p_k\}_{k \geq 0}$ converges almost surely to a compact connected subset of C_n . In particular if \mathcal{E}_n has only isolated critical points in $Q^n \setminus D_n$, the sequence $\{p_k\}_{k \geq 0}$ converges to one of them almost surely.

The proof of Theorem 3 can be found in appendix A. The first part of the theorem is a fairly direct application of Theorem 1, but does not rule out asymptotic convergence to the set D_n of aggregated configurations. This motivates the second part of the theorem.

D. Some Extensions

Before closing this section, we briefly illustrate how the stochastic gradient view leads to simple solutions for interesting variations of the coverage control problem.

1) *A Heterogeneous Coverage Problem:* As in Section 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 , and 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 $\lambda_\alpha, \alpha \in \{a, b, ab\}$, and the agents can observe its type. The spatial distribution \mathbb{P}_α of events of type α is also a priori unknown, and satisfies Assumption 1. Finally, denote the vector of robot positions $p = [p_1^A, \dots, p_{m_A}^A, p_1^B, \dots, p_{m_B}^B]$. The asymptotic configuration of the robots must now optimize the expected cost

$$\begin{aligned} \mathcal{E}_{m_A, m_B}(p) = & \lambda_a E \left[\min_{i \in [m_A]} f_A(\|p_i^A - Z\|) \mid \alpha = a \right] \\ & + \lambda_b E \left[\min_{j \in [m_B]} f_B(\|p_j^B - Z\|) \mid \alpha = b \right] \\ & + \lambda_{ab} E \left[\max \left\{ \min_{i \in [m_A]} f_A(\|p_i^A - Z\|), \right. \right. \\ & \quad \left. \left. \min_{j \in [m_B]} f_B(\|p_j^B - Z\|) \right\} \mid \alpha = ab \right], \end{aligned} \quad (21)$$

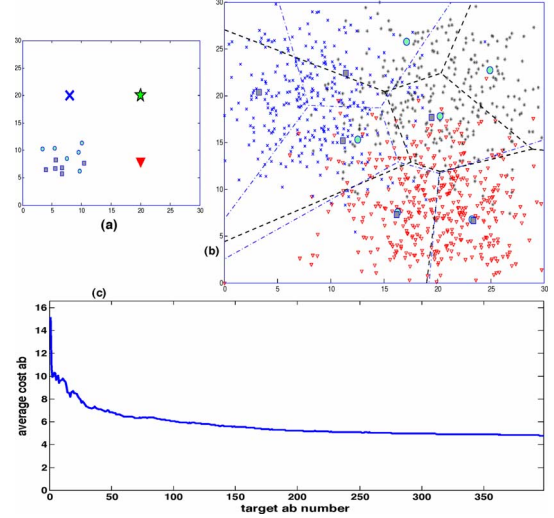


Fig. 1. Heterogeneous coverage control for a system with two types of robots, A (green circles) and B (gray squares). Events requiring service from type a appear with probability 30% and a distribution approximately centered at $[20; 20]^T$ (star on Fig. 1(a)). Targets of type b appear with probability 30% and a distribution approximately centered at $[8; 20]^T$ (cross on Fig. 1(a)). Finally targets of type ab appear with probability 40% and a distribution approximately centered at $[20; 8]^T$ (triangle on Fig. 1(a)). Fig. 1(a) shows the initial robot configuration and Fig. 1(b) 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. 1(c) shows the empirical average cost incurred by the targets of type ab , where the average is taken over all the past targets of this type seen so far.

where f_A and f_B are increasing, continuously differentiable functions with values in $\mathbb{R}_{\geq 0}$. 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. This can model the time necessary for one robot of each type to travel to the event location for example.

For this problem, one can verify as before that the stochastic gradient update rule (10) takes the following surprisingly simple form. When an event of type a appears at location z_{k+1} , the closest robot of type A , say i , services it and changes its reference position by moving it toward z_{k+1} by a (truncated and projected) step $\gamma_k f_A(\|z_{k+1} - p_{i,k}^A\|)((z_{k+1} - p_{i,k}^A)/\|z_{k+1} - p_{i,k}^A\|)$ as in (18), 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 location. Then only this robot moves its reference position by the same step as in (18). 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. 1 for $f_A(x) = f_B(x) = x$.

2) *Target Tracking With Markovian Dynamics:* Suppose 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 a Markov chain with a unique stationary asymptotic 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 (18) to optimize

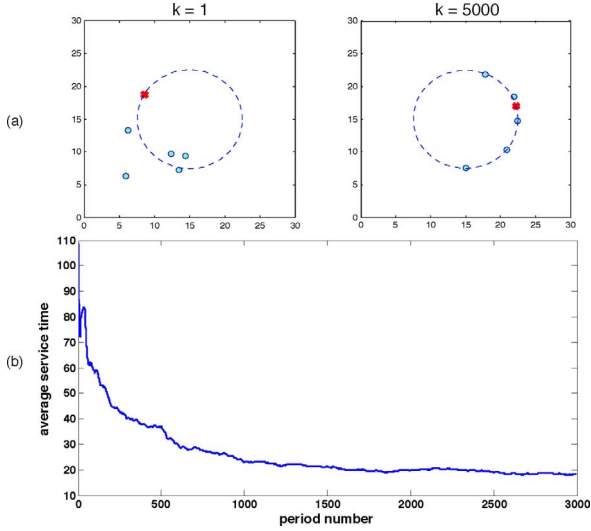


Fig. 2. Adaptive coverage algorithm for a target with Markovian dynamics moving on a circle. Fig. 2(a) shows the positions of the robots (blue circles) and the target (red cross) initially and after 5000 time-steps. The stepsizes used were $\gamma_k = 1/(1 + 5 \times 10^{-3}k)$. The curve on Fig. 2(b) shows the evolution of the empirical average cost over time, where the average is taken over the past 1000 cost measurements.

the steady-state robotic network configuration, and a convergence result similar to Theorem 3 can be proven using stochastic approximation arguments [23, Chapter 1]. This tracking scheme does not require the 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. 2. Although the target distribution does not dominate the Lebesgue measure as required in the second part of Theorem 3, in practice we do not observe convergence to an aggregated configuration. Note how the robots position themselves in the region around the point $[1, 0]^T$ where the target spends most of its time.

IV. ADAPTIVE SPATIAL LOAD-BALANCING AND PARTITIONING

In this section, we design distributed adaptive algorithms that partition the space into n cells, one for each robot, so that the steady-state probability that an event falls into cell i has a pre-specified value a_i . Here we have $a_i > 0$ for all $i \in [n]$, and $\sum_{i=1}^n a_i = 1$. These algorithms allow an operator to specify the steady state utilization of the different robots, by letting each robot service only the events occurring in its cell. Such spatial load balancing algorithms have important applications in multi-robot systems and location optimization, see e.g., [19], [20], [43]. 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 \mathcal{Q} . In this section, \mathcal{Q} is assumed to be compact for simplicity, but not necessarily convex. Based on the observation of the successive event locations, we design a sequence of

partitions of \mathcal{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 $i \neq j$. Our algorithms produce regions whose intersections have \mathbb{P}_z -measure zero, hence this 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 $(k+1)^{\text{th}}$ event. Our sequence of partitions $\{R_{i,k}\}_{i \in [n]}$ converges to a partition $\{R_i\}_{i \in [n]}$, i.e., $d_H(R_{i,k}, R_i) \rightarrow 0$ as $k \rightarrow \infty$, such that $\mathbb{P}_z(R_i) = a_i$ for all $i \in [n]$.

Let $\mathcal{G} = \{g_1, \dots, g_n\}$ be a set of n fixed and distinct 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 $T : \mathcal{Q} \rightarrow \mathcal{G}$, 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. Now consider the following optimization problem

$$\inf_{T \in \mathcal{T}} \int_{\mathcal{Q}} c(z, T(z)) \mathbb{P}_z(dz) \quad (22)$$

$$\text{subject to } \mathbb{P}_z(T^{-1}(g_i)) = a_i, i \in [n], \quad (23)$$

where $c : \mathcal{Q} \times \mathcal{G} \rightarrow \mathbb{R}$ is a given cost function. The following theorem gives a general way of producing assignments or partitions that optimize (22), (23).

Theorem 4: Consider problem (22), (23), where \mathcal{Q} is compact, and assume that

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

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

Then the problem admits an assignment $T \in \mathcal{T}$ that attains the infimum in (22). The value of the optimization problem is equal to

$$\max_{w \in \mathbb{R}^n} h(w) := \int_{\mathcal{Q}} \min_{i \in [n]} \{c(z, g_i) - w_i\} \mathbb{P}_z(dz) + \sum_{i=1}^n a_i w_i, \quad (24)$$

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

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

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

$$[-\mathbb{P}_z(V_1^c(\mathcal{G}, w)) + a_1, \dots, -\mathbb{P}_z(V_n^c(\mathcal{G}, w)) + a_n]^T. \quad (25)$$

Finally, the following supergradient optimization algorithm

$$\begin{aligned} w_0 &= 0, \\ w_{i,k+1} &= w_{i,k} + \gamma_k [-\mathbb{P}_z(V_i^c(\mathcal{G}, w_k)) + a_i], \quad i = 1, \dots, N, \end{aligned} \quad (26)$$

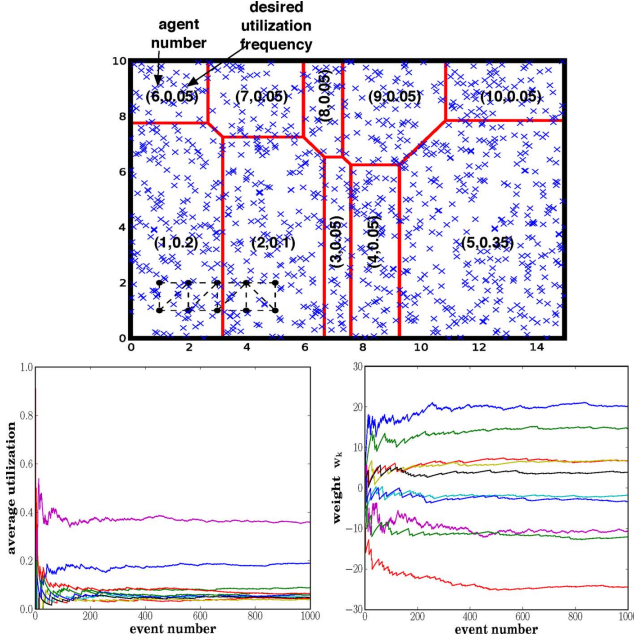


Fig. 3. Partition for 10 robots after 1000 events for the quadratic cost $c(z, g_i) = \|z - g_i\|^2$. The partition at each step is a power diagram. The desired utilization rates are shown for each agent on the figure. The power diagram generators used are represented as black dots in the lower left corner. Note that fixing their positions determines the orientation of the cell boundaries [31]. The power cells shown in red are computed using CGAL [44], but need not be computed by the agents running Algorithm 1. The bottom left figure shows the evolution of the empirical utilization frequencies over the first 1000 events, and the bottom right figure the evolution of the weight vector w_k . The chosen stepsizes were $\gamma_k = 10/(1 + 0.01k)$.

where γ_k is a sequence of stepsizes satisfying (12), converges to an optimal set of weights maximizing h .

In other words, there is a set of weights $w^* \in \mathbb{R}^n$, maximizing the dual function defined in (24), for which the corresponding generalized Voronoi cells $\{V_i^c(\mathcal{G}, w^*)\}_{i \in [n]}$ defined in (3) satisfy the constraints of interest (23). In addition, the assignment corresponding to these regions minimizes the expected cost (22). 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 generalized Voronoi diagram becomes a power diagram. Because the boundaries of the power cells are hyperplanes in \mathbb{R}^q [30], our Assumption A2 on \mathbb{P}_z in Theorem 4 is satisfied under Assumption 1. Theorem 4 generalizes some results in [19], [20], [43] by imposing weaker conditions on \mathbb{P}_z and c . A proof is provided in Appendix B, based on results from optimal transportation [26]–[28].

For our scenario where \mathbb{P}_z is unknown, we replace the gradient ascent algorithm (26) by a stochastic version presented in Algorithm 1, and whose behavior is illustrated on Fig. 3. For simplicity, we specialize the discussion to $c(z, g_i) = f(\|z - g_i\|)$, where f is increasing, and denote the generalized Voronoi cells by $V_i^f(\mathcal{G}, w)$. If, at period k , the event is located at Z_k , a possible choice for the stochastic supergradient is simply

$$[-\mathbf{1}_{\{V_1^f(\mathcal{G}, w_{k-1})\}}(Z_k) + a_1, \dots, -\mathbf{1}_{\{V_n^f(\mathcal{G}, w_{k-1})\}}(Z_k) + a_n]^T. \quad (27)$$

Computing component i of (27) relies on testing if $Z_k \in V_i^f(\mathcal{G}, w_{k-1})$, which is much easier than computing the \mathbb{P}_z -area of the generalized Voronoi cell as in (26). For this

test, 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 Section II-D) with $\hat{d}_i = f(\|Z_k - g_i\|) - w_{i,k-1}$ (and $\hat{d}_i = +\infty$ if agent i did not detect the event). The following result is now a direct application of Theorem 2.

Theorem 5: Assume that the stepsizes γ_k in Algorithm 1 satisfy (12), and that Assumptions A1, A2 of Theorem 4 are satisfied for $c(z, g_i) = f(\|z - g_i\|)$. Then the weights updated by following Algorithm 1 converge almost surely to a maximizer w^* of (24), and the resulting generalized Voronoi diagram $\{V_i^c(\mathcal{G}, w^*)\}_{i \in [n]}$ satisfies the utilization constraints (23).

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 (22).

Initialization: for $i \in [n]$, $w_i \leftarrow 0$.

When event $k \geq 1$ appears at location Z_k :

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

- if robot i terminates FloodMin with $d_i = \hat{d}_i$ then

$$w_i \leftarrow w_i + \gamma_{k-1}(a_i - 1)$$

- else

$$w_i \leftarrow w_i + \gamma_{k-1}a_i.$$

- end if

V. ADAPTIVE DYNAMIC VEHICLE ROUTING

We now combine the algorithms of Sections III-C and 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., $q = 2$. In the DTRP [13], events appear in the workspace \mathcal{Q} according to a space-time Poisson process with rate λ and spatial distribution \mathbb{P}_z . We assume as in Section III-C that \mathcal{Q} is convex and compact. 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 variables S_k are iid with finite first and second moments \bar{s}, \bar{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 $\bar{\Sigma} = \limsup_{k \rightarrow \infty} E[\Sigma_k]$. Let $\rho = \lambda \bar{s} / n$ denote the load factor, i.e., the average fraction of time a robot spends in on-site service [15]. Policies for the DTRP are usually analyzed in two limiting regimes, namely in light traffic conditions ($\lambda \rightarrow 0^+$) and heavy traffic conditions ($\rho \rightarrow 1^-$).

The policies for the DTRP initially proposed in [13]–[15] require the knowledge of the event distribution \mathbb{P}_z . In contrast, [18], [22] propose algorithms for the DTRP that work without this knowledge in the light traffic regime, but leave open the adaptive problem in heavy traffic. The following sections make

two contributions to the DTRP. First, in the light traffic case, we use the adaptive coverage control algorithm of Section III-C to obtain an adaptive policy that is simpler than the solutions proposed so far [18], [22] and provides the same convergence guarantees. Second, for the heavy traffic case, we present the first fully adaptive policy for the DTRP that provably stabilizes the system as long as it is stabilizable, in the absence of knowledge of \mathbb{P}_z . This policy relies on the adaptive partitioning algorithm of Section IV.

A. Light Traffic Regime

Note first that we always have [14]

$$\bar{\Sigma} \geq \min_p \mathcal{E}_n(p) + \bar{s}, \quad (28)$$

where $\mathcal{E}_n(p)$ is defined by (4) for $f(x) = x/v$. This bound is tight in light traffic conditions [13], [15], and achieved by the following policy. Let $p^* = [p_1^*, \dots, p_n^*] \in \mathbb{Q}^n$ 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 $\lambda \rightarrow 0^+$, the agents are at their reference configuration p^* when a new event occurs, and this policy achieves the bound (28)[15].

To obtain an adaptive version of the above policy, we can use the coverage control algorithm of Section III-C to find a local minimizer of \mathcal{E}_n . In the absence of an event, each robot waits at its current 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, updates its reference position to $p_{j,k} = \Pi_Q [p_{j,k-1} + \gamma_k(1/v)((Z_k - p_{j,k-1})/\|Z_k - p_{j,k-1}\|)]$, and travels back toward $p_{j,k}$. Reasoning as in [13], [15], in the light traffic case where $\lambda \rightarrow 0$, the agents are at their reference positions when a new event occurs. Hence 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 (18) converges under the assumptions of Theorem 3. For $n = 1$, it achieves the minimum system time since \mathcal{E}_1 is convex. A similar guarantee is provided by the adaptive light traffic policy described in [22], at the expense of a significantly more complex algorithm where the robots keep track of all past locations visited. Note that these policies turn out to be unstable as the load factor ρ increases [15], which motivates the heavy traffic policy of the next section.

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., [15], [18], [45], [46]. The following non-adaptive policy, although not the best available, stabilizes the system in heavy-traffic, i.e., as $\rho \rightarrow 1^-$ [18], [45], [46]. We partition the workspace \mathbb{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 R_i . It does so by forming successive traveling salesman tours (TSP tours) through the event locations falling in this 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 visiting each point in the set once. While robot i services 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. Assuming that \mathbb{P}_z has a density ϕ_z , it is known that this policy achieves the following bounds on the system time in heavy-traffic [18, theorems 4.2, 6.4]

$$\frac{C^*}{n^2} \leq \lim_{\rho \rightarrow 1^-} (1 - \rho)^2 \bar{\Sigma} \leq \frac{2C^*}{n},$$

$$\text{where } C^* = C \frac{\lambda \left(\int_Q \phi_z(z)^{1/2} dz \right)^2}{v^2} \text{ and } C \approx 0.253. \quad (29)$$

In addition, the right-hand side of (29) can be changed to $2C^*/n^2$ if \mathbb{P}_z is the uniform distribution on \mathbb{Q} [18]. Now consider the adaptive version of this policy described in Algorithm 2, which partitions the workspace as in Section IV, and works without the knowledge of any event process parameter such as λ or \mathbb{P}_z .

Theorem 6: *The adaptive policy of Algorithm 2 stabilizes the system as long as $\rho < 1$ and achieves a steady-state system time satisfying the heavy traffic performance bound (29). Moreover if $n = 1$, this adaptive policy is also optimal in the light traffic regime $\lambda \rightarrow 0^+$.*

Proof: As $\rho \rightarrow 1$, the region of each robot is never empty and hence the robot never enters the mode in Algorithm 2 where it goes toward its reference position p_i [18]. By Theorem 5, the space partition allocating events to robots converges as $k \rightarrow \infty$ 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 (29). In the light traffic regime ($\lambda \rightarrow 0$) and in steady-state, each agent following Algorithm 2 is at the median of its region R_i when a new event occurs. Indeed the updates of the reference position p_i can be viewed as a stochastic gradient descent algorithm for the cost $g_i(p_i) = E[\|p_i - Z\| | Z \in R_i]$ (notice here that the space partition evolves independently of the reference locations $p_i, i \in [n]$, in contrast to Section III-C). In particular if $n = 1$, there is just one cell and \mathcal{E}_1 is strictly convex, so the policy achieves the performance bound (28). ■

Algorithm 2 Adaptive DTRP algorithm. Robot i updates a weight $w_i \in \mathbb{R}$ as in **Section IV**, a reference position $p_i \in \mathbb{R}^q$, and two sets of event locations \mathcal{O}_i and \mathcal{P}_i . It is also associated to a fixed point $g_i \in \mathbb{Q}$, with $g_i \neq g_j$, for $i \neq j$.

Initialization: for $i \in [n]$, $w_i \leftarrow 0$, $p_i \leftarrow$ robot i 's initial position, $\mathcal{O}_i \leftarrow \emptyset$, $\mathcal{P}_i \leftarrow \emptyset$.

When event $k \geq 1$ appears at location Z_k :

- Run the FloodMin algorithm starting with $\hat{d}_j = \|Z_k - g_j\|^2 - w_j, j \in [n]$.
- if robot i terminates FloodMin with $d_i = \hat{d}_i$ then

$w_i \leftarrow w_i + \gamma_{k-1}(n-1)/n$,
 $p_i \leftarrow \Pi_Q [p_i + \gamma_{k-1}((Z_k - p_i)/\|Z_k - p_i\|)]$,
 $\mathcal{O}_i \leftarrow \mathcal{O}_i \cup \{Z_k\}$.

- else

$w_i \leftarrow w_i + \gamma_{k-1}/n$, and p_i, \mathcal{O}_i remain unchanged.

- end if

In parallel, execute the following process forever for each robot $i \in [n]$

- 1) When $\mathcal{P}_i = \mathcal{O}_i = \emptyset$, robot i travels toward p_i and stays there if p_i is reached.
- 2) When $\mathcal{P}_i = \emptyset$ and $\mathcal{O}_i \neq \emptyset$, then let $\mathcal{P}_i \leftarrow \mathcal{O}_i$, $\mathcal{O}_i \leftarrow \emptyset$. Compute an Euclidean TSP tour through the points \mathcal{P}_i .
- 3) When $\mathcal{P}_i \neq \emptyset$, then service the locations in \mathcal{P}_i in the order of the TSP tour, removing them from \mathcal{P}_i when they are serviced. At the end of the tour, we are back in the situation $\mathcal{P}_i = \emptyset$. If $\mathcal{O}_i = \emptyset$, go to 1), otherwise, go to 2).

VI. CONCLUSIONS

We have discussed robot deployment algorithms for coverage control, spatial partitioning and dynamic vehicle routing problems in the situation where the event location 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 problem). 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., [37], [39]. 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. Once the robots have converged, the adaptive algorithm is used to correct for the modeling errors and environmental uncertainty, exploiting actual observations. Note also that the stochastic gradient algorithms can still be used if the distribution \mathbb{P}_z is known, by generating random targets artificially and essentially evaluating integrals such as (5) by Monte-Carlo simulations [33]. However, this method is generally only advantageous for dimensions q sufficiently large.

APPENDIX A

CONVERGENCE OF THE COVERAGE CONTROL ALGORITHM

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

$$\dot{p} = -\nabla \mathcal{E}_n(p), \quad p(0) \in \mathbb{Q}^n \setminus D_n, \quad (30)$$

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

$$\dot{p} \in \mathcal{H}_n(p), \quad p(0) \in \mathbb{Q}^n, \quad (31)$$

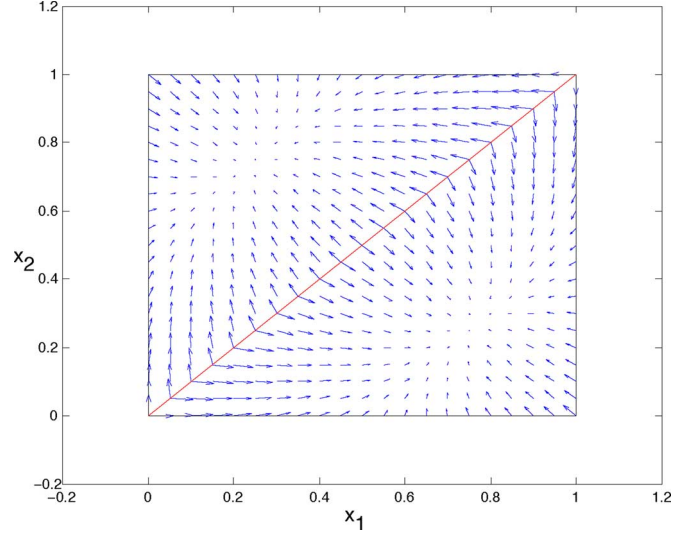


Fig. 4. Vector field for the gradient system (30), with two agents evolving on $[0, 1]$ and \mathbb{P}_z uniform on $[0, 1]$. The discontinuity on the line $x_1 = x_2$ occurs when the two agents switch side, from $x_1 < x_2$ to $x_1 > x_2$. Note that the vector field is symmetric with respect to this 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 robots. The differential inclusion (31) has an additional (unstable) equilibrium at $(1/2, 1/2)$.

where the set-valued map \mathcal{H}_n is defined in (20). Note that $\nabla \mathcal{E}_n$ is piecewise continuous and \mathcal{H}_n can in fact equivalently be defined as [29, p.51]

$$\mathcal{H}_n(p) = \begin{cases} \{-\nabla \mathcal{E}_n(p)\} & \text{if } p \notin D_n, \\ \overline{\text{co}} \{ \lim_{k \rightarrow \infty} (-\nabla \mathcal{E}_n(p_k)) \mid p_k \rightarrow p \text{ as } k \rightarrow \infty \} & \text{if } p \in D_n. \end{cases} \quad (32)$$

Following the ODE method [36], we can characterize the asymptotic behavior of the algorithm (18) as in Theorems 1 and 3 by studying the properties of these continuous-time dynamical systems. We assume as in Section III-C that $f : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ is increasing and continuously differentiable. We refer the reader to [8], [30], [33], [47] for previous work on the gradient system (30). In particular, [33] provides some convergence results for algorithm (18), and points out that the non-differentiability of \mathcal{E}_n creates technical difficulties in the convergence proofs. We handle these issues by initially considering the differential inclusion (31) instead of the ODE (30). 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 [34].

Differentiability Properties of \mathcal{E}_n : Recall from Proposition 1 that \mathcal{E}_n is continuously differentiable on $\mathbb{R}^n \setminus D_n$. In general however, $\nabla \mathcal{E}_n$ is discontinuous on the set D_n , see Fig. 4. To discuss more precisely the behavior of the gradient of \mathcal{E}_n as we approach the set D_n , define

$$N(x) = \begin{cases} \|\nabla \mathcal{E}_n(x)\|^2 & \text{if } x \in \mathbb{R}^n \setminus D_n \\ \liminf_{y \in \mathbb{R}^n \setminus D_n, y \rightarrow x} \|\nabla \mathcal{E}_n(y)\|^2 & \text{if } x \in D_n. \end{cases}$$

Note that because $\nabla \mathcal{E}_n$ is continuous on $\mathbb{R}^n \setminus D_n$ the two definitions of N in fact coincide on this set. The proof of the next proposition follows that of [33, Lemma 30].

Proposition 2: Suppose that Assumption 1 holds and that \mathbb{P}_z dominates the Lebesgue measure on Q^n . Then we have $N(p) > 0$ for all $p \in D_n \cap Q^n$. Hence there exists $\delta_0 > 0$ such that

$$\inf_{p \in Q^n \cap (B(D_n \cap Q^n, \delta_0) \setminus D_n)} \|\nabla \mathcal{E}_n(p)\|^2 =: \kappa > 0.$$

Trajectories of the Gradient System: We now turn to the study of the trajectories of the ODE (30) and the differential inclusion (31). The following general result follows from arguments as in [33, lemma 33], see also [34].

Proposition 3: Suppose that Assumption 1 holds and that \mathbb{P}_z dominates the Lebesgue measure on Q^n . If $x_0 \in Q^n \setminus D_n$, a trajectory $t \rightarrow x(t)$ of the ODE (30) with $x(0) = x_0$ remains in $Q^n \setminus D_n$, i.e., for all $t < \infty$, $x(t) \in Q^n \setminus D_n$. Moreover, it converges to a compact connected subset of $\{x \in Q^n \setminus D_n : \nabla \mathcal{E}_n(x) = 0\}$. Finally, a trajectory of the differential inclusion (31) starting from $x_0 \in Q^n$ remains in Q^n .

Next, we show that the trajectories of the ODE never stay in $B(D_n \cap Q^n, \delta_0)$ for a long time.

Corollary 1: Suppose that Assumption 1 holds and that \mathbb{P}_z dominates the Lebesgue measure on Q^n . Let $\delta_0 > 0, \kappa > 0$ be defined as in Proposition 2, $x_0 \in Q^n \cap (B(D_n \cap Q^n, \delta_0) \setminus D_n)$, and let $T = \max_{x \in Q^n \cap B(D_n \cap Q^n, \delta_0)} \mathcal{E}_n(x)/\kappa$. Then a trajectory of the ODE passing through x_0 at time t_1 must exit $B(D_n \cap Q^n, \delta_0)$ at some time $t_2 \leq t_1 + T$.

Proof: We have, for $t \geq t_1$ and as long as the trajectory $t \rightarrow x(t)$ remains in $B(D_n \cap Q^n, \delta_0) \setminus D_n$

$$\begin{aligned} 0 \leq \mathcal{E}_n(x(t)) &= \mathcal{E}_n(x_0) - \int_{t_1}^t \|\nabla \mathcal{E}_n(x(s))\|^2 ds \\ &\leq \max_{x \in Q^n \cap (B(D_n \cap Q^n, \delta_0) \setminus D_n)} \mathcal{E}_n(x) - \kappa(t - t_1). \end{aligned}$$

Hence the trajectory must exit $B(D_n \cap Q^n, \delta_0) \setminus D_n$ 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 $B(D_n \cap Q^n, \delta_0)$. ■

The set C_n defined in (19) contains the set of limit points of the ODE (30) by Proposition 3. From the definition of the set-valued map \mathcal{H}_n , the set \mathcal{L} of limit points of the differential inclusion (31) consists of the set of limit points of the ODE (30) 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 $\mathcal{L} \subset C_n \cup (D_n \cap Q^n)$. Moreover, we know by Proposition 2 that $C_n \subset Q^n \setminus B(D_n, \delta_0)$ if \mathbb{P}_z dominates the Lebesgue measure.

Convergence of the Adaptive Coverage Control Algorithm: We now prove the main convergence theorem for the adaptive coverage control algorithm.

Proof of Theorem 3: First, for the proof of convergence, we can ignore the projection Π_Q in (18). In general, the analysis involves the corresponding projected ODE or projected differential inclusion, see [38], [39, chapter 5]. Note however from Proposition 3 that at any boundary point of Q^n , the velocity vector of the unprojected differential inclusion is already in the tangent cone of Q^n . Hence the projection step does not change the continuous-time dynamics and the convergence properties

remain the same as for the unprojected algorithm. Moreover, the saturation function does not change the convergence properties either [39, Section 1.3.5].

Now the fact that with probability one, a sequence of iterates of (18) converges to a compact connected invariant set of the differential inclusion (31) is standard, see e.g., [38, chapter 5], [39, Theorem 8.1 p. 195]. Consider a sample ω such that $\{p_k(\omega)\}$ converges to such a set, denoted S . In view of Proposition 3, we have $S \subset Q^n$. Suppose that S is not entirely contained in D_n , and take $a \in S \setminus D_n$. Then a trajectory of the differential inclusion passing through a at $t = 0$ is in fact a trajectory of the ODE (30) for $t \geq 0$, 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., $a \in C_n$. This proves the first part of the theorem.

If \mathbb{P}_z dominates the Lebesgue measure, then we know that C_n and D_n are disconnected by Proposition 2, so S is contained in one of these sets. Recall that under Assumption 1, we can assume that almost surely, the iterates $\{p_k\}_{k \geq 0}$ of (18) never hit D_n . Choose the sample ω above in this set of probability 1, and recall the definitions of δ_0 and T from Corollary 1. Suppose now that $S \subset D_n$. Then there exists k_0 such that for all $k \geq k_0$, $p_k \in B(D_n, \delta_0/4)$. For any $k \geq 0$, denote by $x^k(\cdot)$ the solution of the ODE (30) 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 [38, 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 must have $d(x^k(t_k + T), D_n) > \delta_0$. 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 $d(p_{\tilde{k}}, D_n) > \delta_0/2$. But this contradicts our assumptions that $p_{\tilde{k}} \in B(D_n, \delta_0/4)$. Hence we cannot have $S \subset D_n$ and so $S \subset C_n$. This finishes the proof of the theorem. ■

APPENDIX B

SPACE PARTITIONING AND OPTIMAL TRANSPORTATION

In this section we prove Theorem 4, 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 [8], [19], this theorem makes weaker assumptions on the cost function $c(z, g)$ and on the target distribution \mathbb{P}_z . The main tool on which Theorem 4 relies is Kantorovich duality [26]. See also [27], [48], [49] for related results.

Proof of Theorem 4: We start by relaxing the optimization problem (22), (23) to the following Monge-Kantorovich optimal transportation Problem (MKP) [26]. Let $P_2 = \sum_{i=1}^n a_i \delta_{g_i}$, so that (23) can be rewritten $\mathbb{P}_z \circ T^{-1} = P_2$. We consider the minimization problem

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

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

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

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 $c(z, g)$ for moving a unit of mass from z to g . 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 , is a Monge Problem (MP) [50], and is exactly our problem (22), (23). 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, and assuming the infimum in (22) is attained. This is the case if c is lower bounded and lower semicontinuous and \mathbb{P}_z is tight [26, Remark 2.1.2], and this last condition is satisfied since we assume Q compact. Next, by Kantorovitch duality [26], we have

$$\begin{aligned} \min_{\pi \in \mathcal{M}(\mathbb{P}_z, P_2)} \int_{Q \times Q} c(z, g) d\pi(z, g) \\ = \sup_{(\phi, w) \in \Phi_c} \left\{ \int_Q \phi(z) d\mathbb{P}_z(z) + \sum_{i=1}^n a_i w_i \right\}, \end{aligned} \quad (33)$$

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

$$\phi(z) + w_i \leq c(z, g_i), \quad (34)$$

for \mathbb{P}_z -almost all z in Q and for all i in $[n]$. Now for any $w \in \mathbb{R}^n$, define the function $w^c : Q \rightarrow \mathbb{R}$ 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 (33) to pairs of the form (w^c, w) . Combining this with the previous remark on the Monge solution to the Monge-Kantorovitch problem, we get

$$\begin{aligned} \min_{\substack{T: Q \rightarrow \{g_1, \dots, g_n\} \\ \mathbb{P}_z \circ T^{-1} = P_2}} \int_Q c(z, T(z)) \mathbb{P}_z(dz) \\ = \sup_{w \in \mathbb{R}^n} \left\{ \int_Q \min_{i \in [n]} \{c(z, g_i) - w_i\} \mathbb{P}_z(dz) + \sum_{i=1}^n a_i w_i \right\}. \end{aligned} \quad (35)$$

Hence the value of the optimization problem is equal to the supremum of the function h defined in (24). The fact that the supremum is attained in the right hand side of (35) follows from, e.g., [26, Theorem 2.3.12] under our assumption A1 for c .

The function h is concave since $w \rightarrow \min_{i \in [n]} \{c(z, g_i) - w_i\}$ 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

$$\begin{aligned} h(w^2) - h(w^1) &= \int_Q \min_{i \in [n]} \{c(z, g_i) - w_i^2\} \mathbb{P}_z(dz) \\ &\quad - \int_Q \min_{i \in [n]} \{c(z, g_i) - w_i^1\} \mathbb{P}_z(dz) + \sum_{i=1}^n a_i (w_i^2 - w_i^1). \end{aligned}$$

Denoting T^1 an assignment that is optimal for w^1 (given by a generalized Voronoi partition), we have then, for all $z \in Q$,

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

and so

$$\begin{aligned} h(w^2) - h(w^1) &\leq - \sum_{i=1}^n \mathbb{P}_z(V_i^c(\mathcal{G}, w^1)) (w_i^2 - w_i^1) \\ &\quad + \sum_{i=1}^n a_i (w_i^2 - w_i^1). \end{aligned}$$

But this inequality exactly says that $[a_1 - \mathbb{P}_z(V_1^c(\mathcal{G}, w^1)), \dots, a_n - \mathbb{P}_z(V_n^c(\mathcal{G}, w^1))]^T$ is a supergradient of h at w^1 . For the convergence of the supergradient algorithm, see [40, Proposition 8.2.6. p. 480]. ■

REFERENCES

- [1] A. Howard, M. J. Mataric, and G. Sukhatme, "Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem," in *Proc. 6th Int. Symp. DARS*, Fukuoka, Japan, Jun. 2002.
- [2] P. Ögren, E. Fiorelli, and N. Leonard, "Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment," *IEEE Trans. Autom. Control*, vol. 49, no. 8, pp. 1292–1302, Aug. 2004.
- [3] F. Bullo, J. Cortés, and S. Martinez, *Distributed Control of Robotic Networks*. Princeton: Princeton Univ. Press, 2009.
- [4] J. Cortés, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Trans. Robot. Automation*, vol. 20, no. 2, pp. 243–255, Apr. 2004.
- [5] R. Gray, "Vector quantization," *IEEE ASSP Mag.*, vol. 1, no. 2, pp. 4–29, Apr. 1984.
- [6] Z. Drezner, Ed., *Facility Location: A Survey of Applications and Methods*. New York, Springer-Verlag, 1995, Springer Series in Operations Research.
- [7] S. Lloyd, "Least-squares quantization in PCM," *IEEE Trans. Inform. Theory*, vol. 28, no. 2, pp. 129–137, Mar. 1982.
- [8] J. Cortés, S. Martinez, 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, Oct. 2005.
- [9] W. Li and C. G. Cassandras, "Distributed cooperative control of sensor networks," in *Proc. Conf. Decision and Control*, Seville, Spain, Dec. 2005, pp. 2542–2547.
- [10] K. Guruprasad and D. Ghose, "Generalized Voronoi Partition based Multi-Agent Search using Heterogeneous Sensors," Tech. Rep. Indian Institute of Science, Bangalore, 2009 [Online]. Available: <http://arxiv.org/abs/0908.2683v1>
- [11] L. Pimenta, V. Kumar, R. Mesquita, and G. Pereira, "Sensing and coverage for a network of heterogeneous robots," in *Proc. Conf. Decision and Control*, Cancun, Mexico, Dec. 2008, pp. 3947–3952.
- [12] M. Schwager, D. Rus, and J.-J. Slotine, "Decentralized, adaptive coverage control for networked robots," *Int. J. Robotics Research*, vol. 28, no. 3, pp. 357–375, Mar. 2009.
- [13] D. J. 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, Jul.-Aug. 1991.
- [14] D. J. Bertsimas and G. J. Van Ryzin, "Stochastic and dynamic vehicle routing in the Euclidean plane with multiple capacitated vehicles," *Operations Research*, vol. 41, no. 1, pp. 60–76, Jan.-Feb. 1993.
- [15] D. J. Bertsimas and G. J. Van Ryzin, "Stochastic and dynamic vehicle routing with general demand and interarrival time distributions," *Adv. Appl. Probability*, vol. 25, no. 4, pp. 947–978, Dec. 1993.
- [16] E. Frazzoli and F. Bullo, "Decentralized algorithms for vehicle routing in a stochastic time-varying environment," in *Proc. Conf. Decision and Control*, Paradise Island, The Bahamas, Dec. 2004, pp. 3357–3363.
- [17] K. Savla, "Multi-UAV Systems with Motion and Communication Constraints," Ph.D. dissertation, Univ. California, Santa Barbara, 2007.

- [18] M. Pavone, E. Frazzoli, and F. Bullo, "Adaptive and distributed algorithms for vehicle routing in a stochastic and dynamic environment," *IEEE Trans. Autom. Control*, vol. 56, no. 6, pp. 1259–1274, Jun. 2011.
- [19] M. Pavone, A. Arsie, E. Frazzoli, and F. Bullo, "Distributed algorithms for environment partitioning in mobile robotic networks," *IEEE Trans. Autom. Control*, vol. 56, no. 8, pp. 1834–1848, Aug. 2011.
- [20] J. Cortés, "Coverage control and spatial load balancing by robotic sensor networks," *IEEE Trans. Autom. Control*, vol. 55, no. 3, pp. 749–754, Mar. 2010.
- [21] J. Choi and R. Horowitz, "Learning coverage control of mobile sensing agents in one-dimensional stochastic environments," *IEEE Trans. Autom. Control*, vol. 55, no. 3, pp. 804–809, Mar. 2010.
- [22] A. Arsie, K. Savla, and E. Frazzoli, "Efficient routing algorithms for multiple vehicles with no explicit communications," *IEEE Trans. Autom. Control*, vol. 54, no. 10, pp. 2302–2317, Oct. 2009.
- [23] A. Benveniste, M. Metivier, and P. Priouret, *Adaptive Algorithms and Stochastic Approximations*. Berlin: Springer, 1990.
- [24] J. Le Ny and G. J. Pappas, "Adaptive Robot Deployment Algorithms," Tech. Rep. University of Pennsylvania, 2010 [Online]. Available: http://repository.upenn.edu/ese_reports/4/
- [25] J. Le Ny, A. Ribeiro, and G. J. Pappas, "Adaptive communication-constrained deployment of unmanned vehicle systems," *IEEE J. Selected Areas in Commun.*, vol. 30, no. 5, pp. 923–934, Jun. 2012.
- [26] S. T. Rachev and L. Rüschendorf, *Mass Transportation Problems*. Berlin: Springer-Verlag, 1998, vol. 1, Theory.
- [27] W. Gangbo and R. McCann, "The geometry of optimal transportation," *Acta Mathematica*, vol. 177, no. 2, pp. 113–161, 1996.
- [28] L. Rüschendorf and L. Uckelmann, "On optimal multivariate couplings," in *Distributions with Given Marginals and Moment Problems*, V. Benes and J. Stepán, Eds. Norwell, MA: Kluwer Academic, 1997, pp. 261–273.
- [29] J. Cortés, "Discontinuous dynamical systems – A tutorial on solutions, nonsmooth analysis, and stability," *IEEE Control Syst. Mag.*, vol. 28, no. 3, pp. 36–73, Jun. 2008.
- [30] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. Hoboken: Wiley, 2000.
- [31] F. Aurenhammer, "Voronoi diagrams – A survey of a fundamental geometric data structure," *ACM Computing Surveys*, vol. 23, no. 3, pp. 345–405, Sep. 1991.
- [32] N. A. Lynch, *Distributed Algorithms*. Burlington, MA: Morgan Kaufmann, 1996.
- [33] G. Pagès, "A space quantization method for numerical integration," *J. Computational and Appl. Mathemat.*, vol. 89, no. 1, pp. 1–38, Mar. 1997.
- [34] J. Le Ny and G. J. Pappas, "Adaptive Algorithms for Coverage Control and Space Partitioning in Mobile Robotic Networks," Tech. Rep. Univ. Pennsylvania, 2010 [Online]. Available: <http://arxiv.org/abs/1011.0520>
- [35] S. Fekete, J. Mitchell, and K. Beurer, "On the continuous Fermat-Weber problem," *Operations Research*, vol. 53, no. 1, pp. 61–76, Jan.-Feb. 2005.
- [36] L. Ljung, "Analysis of recursive stochastic algorithms," *IEEE Trans. Autom. Control*, vol. 22, no. 4, pp. 551–575, Aug. 1977.
- [37] J. C. Spall, *Introduction to Stochastic Search and Optimization*. Hoboken, NJ: Wiley, 2003.
- [38] V. Borkar, *Stochastic Approximation: A Dynamical Systems Viewpoint*. Cambridge: Cambridge Univ. Press, 2008.
- [39] H. J. Kushner and G. G. Yin, *Stochastic Approximation and Recursive Algorithms and Applications*, 2nd ed. Berlin: Springer, 2003.
- [40] D. P. Bertsekas, A. Nedić, and A. E. Ozdaglar, *Convex Analysis and Optimization*. Belmont, MA: Athena Scientific, 2003.
- [41] J. Mac Queen, L. Le Cam and J. Neyman, Eds., "Some methods for the classification and analysis of multivariate observations," in *Proc. Fifth Berkeley Symp. Mathemat. Statistics and Probability*, 1967, pp. 281–297.
- [42] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybern.*, vol. 43, pp. 59–69, 1982.
- [43] F. Aurenhammer, F. Hoffmann, and B. Aronov, "Minkowski-type theorems and least-squares clustering," *Algorithmica*, vol. 20, pp. 61–76, 1998.
- [44] CGAL, *Computational Geometry Algorithms Library*, [Online]. Available: <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," *Eur. J. Operations Research*, vol. 95, no. 1, pp. 167–177, Nov. 1996.
- [47] Q. Du, V. Faber, and M. Gunzburger, "Centroidal Voronoi tessellations: Applications and algorithms," *SIAM Review*, vol. 41, no. 4, pp. 637–676, Dec. 1999.
- [48] T. Abdellaoui, "Distances de Deux Lois Dans les Espaces de Banach," Ph.D. dissertation, Université de Rouen, Rouen, 1993.
- [49] L. Rüschendorf and L. Uckelmann, "Numerical and analytical results for the transportation problem of Monge-Kantorovich," *Metrika*, vol. 51, no. 3, pp. 245–258, 2000.
- [50] G. Monge, "Mémoire sur la théorie des déblais et des remblais," in *Histoire de l'Académie Royale des Sciences de Paris, avec les Mémoires de Mathématique et de Physique pour la même année*, 1781, pp. 666–704.
- [51] J. A. Cuesta-Albertos and A. Tuero-Díaz, "A characterization for the solution of the Monge-Kantorovich mass transference problem," *Statistics and Probability Lett.*, vol. 16, no. 2, pp. 147–152, Jan. 1993.



Jerome Le Ny (S'05-M'09) received the B.S. degree from the Ecole Polytechnique, Ecole Polytechnique, Palaiseau, France, in 2001, the M.Sc. degree in Electrical Engineering from the University of Michigan, Ann Arbor, in 2003, and the Ph.D. degree in Aeronautics and Astronautics from the Massachusetts Institute of Technology, Cambridge, in 2008.

He is an Assistant Professor in the Department of Electrical Engineering at the École Polytechnique de Montréal, Canada, since May 2012. From 2008 to 2012 he was a Postdoctoral Researcher with the GRASP Laboratory at the University of Pennsylvania, Philadelphia. His research interests include robust and stochastic control with applications to autonomous and embedded systems, air traffic control, and the associated computational methods supporting the design of complex systems.



George J. Pappas (S'90-M'91-SM'04-F'09) received the Ph.D. degree in electrical engineering and computer sciences from the University of California, Berkeley in 1998, for which he received the Elisha Jury Award for Excellence in Systems Research.

He is currently the Joseph Moore Professor of Electrical and Systems Engineering at the University of Pennsylvania, Philadelphia. He is a member of the General Robotics, Automation, Sensing and Perception (GRASP) Laboratory and the PRECISE Center for Embedded Systems. His current research

interests include hybrid and embedded systems, hierarchical control systems, distributed control systems, nonlinear control systems, with applications to robotics, unmanned aerial vehicles, biomolecular networks, and green buildings.

Dr. Pappas has received numerous awards, including the National Science Foundation (NSF) CAREER Award in 2002, the NSF Presidential Early Career Award for Scientists and Engineers in 2002, the 2009 George S. Axelby Outstanding Paper Award, and the 2010 Antonio Ruberti Outstanding Young Researcher Prize.