Sensor-Based Robot Deployment Algorithms

Jerome Le Ny and George J. Pappas

Abstract— In robot deployment problems, the fundamental issue is to optimize a steady state performance measure that depends on the spatial configuration of a group of robots. For such problems, a classical way of designing high-level feedback motion planners is to implement a gradient descent scheme on a suitably chosen objective function. This can lead to computationally expensive algorithms that may not be adaptive to uncertain dynamic environments. We address these challenges by showing that algorithms for a variety of deployment scenarios in uncertain stochastic environments and with noisy sensor measurements can be designed as stochastic gradient descent algorithms, and their convergence properties analyzed via the theory of stochastic approximations. This approach yields often surprisingly simple algorithms that can accommodate complicated objective functions, and work without a detailed model of the environment. To illustrate the richness of the framework, we discuss two applications, namely source seeking with realistic stochastic wireless connectivity constraints, and coverage with heterogeneous sensors.

I. INTRODUCTION

There has been in the last few years significant research efforts dedicated to the deployment of mobile robotic networks. These systems can be used in a variety of surveillance, monitoring and search applications as reconfigurable sensor networks, concentrating their information gathering activities where it is most critical [1]–[3]. In this paper, we define deployment algorithms as algorithms that aim at reaching a desired steady-state configuration for a robot or group of robots rather than optimizing a trajectory-dependent performance objective. The algorithms considered here follow the same idea as the classical potential function methods for feedback motion planning [4], but are typically implemented at the higher levels of a robot motion planner, where we assume that the robot dynamics can be neglected. The principle underlying these algorithms is to express the goal configuration for the robots as the minimum of a suitably chosen objective function, and to interpret a gradient descent on that function as a motion plan from the initial to the goal configuration. An additional

benefit of such gradient descent algorithms is that they can in fact adapt the configuration to slow or infrequent changes in the environment.

We note that a significant part of the work related to multi-robot deployment relies on such gradient vector fields. This includes formation control and flocking [5]–[9], coverage [2], [10] as well as certain vehicle routing problems [11], or foraging and source seeking [1], [12]. Most of this work assumes a deterministic or at least known model of the environment and neglects various sources of uncertainty that can complicate implementations and invalidate convergence guarantees. More recently, there has been some interest in stochastic deployment scenarios in partially unknown environments with possibly noisy measurements [1], [13]–[16]. An essential idea of this paper is that most of these stochastic deployment problems can be discussed from the unifying point of view of stochastic gradient descent algorithms, thereby simplifying the convergence proofs and allowing to easily derive new algorithms for more complex problems.

The rest of the paper is organized as follows. In section II we review deterministic gradient descent methods using potential fields for static deployment problems. For illustration purposes, we consider two important examples of deployment problems, namely deployment with realistic wireless connectivity constraints, and coverage scenarios, and we point out some limitations of deterministic gradient descent algorithms. Section III recalls basic facts about stochastic approximations and stochastic gradient descent algorithms. Finally, in section IV we revisit the scenarios of section II and illustrate how stochastic gradient algorithms can form the basis of new algorithms for complex deployment problems in the presence of various sources of uncertainty and in the absence of a precise environment model. Convergence proofs are only briefly sketched due to space constraints and generally follow from classical results on stochastic approximations [17]–[19].

II. LIMITATIONS OF DETERMINISTIC DEPLOYMENT ALGORITHMS

In deployment problems we want to drive a mobile robotic network to a fixed steady-state configuration

that optimizes some performance criterion. This criterion does not capture how the robots reach the goal configuration, i.e., transient characteristics such as convergence speed are only analyzed a posteriori for a given scheme. Because transient behavior is not accounted for directly in the performance criterion, numerous strategies can be used to drive the robots to the final configuration of interest. A common technique is to use low level controllers and fast internal feedback loops to present to the high level motion planner an abstract robot model which is fully actuated and has no dynamics, see e.g. the discussions in [4], [10]. Assuming this is feasible, as is done in this paper, we work with the following model. We assume that we have m robots with configurations $p_k = [p_{1,k}, \ldots, p_{m,k}]$ at time $k \in \mathbb{Z}_{\geq 0}$, evolving in a shared environment or workspace Q, i.e. $p_{i,k} \in Q$, for $i = 1, \ldots, m$. At the high-level planning stage we assume that we can work with a fully actuated model

$$p_{i,k+1} = p_{i,k} + u_{i,k}, \;\; i = 1, \ldots, m, \; k \in \mathbb{Z}_{\geq 0}, \quad (1)$$

where u_i is an available control input for robot i. We have velocity constraints of the form $\|u_{i,k}\| \leq \overline{u}_i$, for some $\overline{u}_i \in \mathbb{R}$.

Once a model of the form (1) is assumed, we describe the desired deployment configuration as the minimum of a suitable objective function $f : Q^m \to \mathbb{R}$, also called a potential function, which depends on the configurations p_i of the robots. We can then design an iterative optimization scheme of the gradient descent type to find an (often locally) optimal final configuration, and reinterpret it as a motion for the robots. Namely, we choose the control law

$$u_{i,k} = -\gamma_k \frac{\partial f}{\partial p_i}|_{p=p_k}, \quad (2)$$

for robot i in (1), where $\partial/\partial p_i$ represents the vector of derivatives with respect to the components of p_i, and γ_k is some small, in general time-varying stepsize. These stepsizes can also be used to enforce the velocity constraints most of the time, and we simply truncate $u_{i,k}$ otherwise.

Several issues limit the applicability of such gradient descent schemes however. As the next examples illustrate, in many multi-robot deployment problems, the computation of the gradient in (2) often requires the knowledge of certain a priori unknown environment parameters, or can only be done approximately due to sensor and environment noise, or can be simply too difficult on small platforms with limited computational power. We show in section IV that in many cases these issues can be solved in an elegant way by replacing the deterministic scheme by stochastic gradient algorithms,

which provably work with the very rough approximations of control law (2) arising in practice.

A. Deployment Under Wireless Connectivity Constraints

Communication between robots and operator stations is performed over wireless links and this aspect must be accounted for in deployment problems. Consider the following scenario. A robot must approach a target point q^* in the environment $Q \subset \mathbb{R}^2$, however communication between the robot and the base at $[0,0]^T$ must be maintained at all times. For example, all applications involving Unmanned Aerial Vehicles (UAVs) currently prohibit the loss of communication with any vehicle. Suppose that q^* is outside of the communication range of the base. Then a string of robots can be deployed in order to establish an ad-hoc communication network reaching the target point, see Fig. 1.

Most papers considering such deployment problems use simplified models of the wireless links, typically assuming a deterministic and known function predicting the connectivity at all points of the environment, see e.g. [20] and the references therein. In fact the Signal-to-Noise Ratio (SNR) between a transmitter at p_1 and a receiver at p_2 in Q depends on path loss, shadowing, multipath fading, and the receiver noise power [21]. It is in general a random time-varying quantity, denoted hereafter $SNR_k(p_1, p_2)$ for the period k. Wireless models usually take the form $\log SNR_k(p_1, p_2) = h(p_1, p_2) + \nu_k$, where h is a deterministic quantity capturing path loss, and ν_k is a stochastic zero-mean variation due to shadowing (random effects due to environmental changes) and possibly multipath fading. In [22] the authors consider motion planning problems assuming a realistic communication model but assume an a priori known SNR map, i.e., h and the distribution of ν_k are given. They find that mismatches between the assumed SNR map and the real one have a significant impact on the connectivity. In section IV-A we present an approach that can adapt to an unknown SNR map.

We assume that the random variables ν_k have a steady-state distribution as $k \to \infty$, and for simplicity, we assume that this distribution is independent of p_1, p_2. The following simple potential penalizes points p_1 and p_2 of Q for which the SNR at time k is less than some threshold $\mathtt{SNR_{min}}$

$$c(p_1, p_2) = \begin{cases} \frac{1}{2} \left(-\log SNR_k(p_1, p_2) + \log \mathtt{SNR_{min}} \right)^2 \\ \qquad \text{if } SNR_k(p_1, p_2) < \mathtt{SNR_{min}}, \\ 0 \qquad \text{otherwise.} \end{cases}$$

Now suppose that we look for a simple linear chain configuration, where each robot relays communications

between the robot behind and in front of him, and the last robot m tries to approach the target. Minimizing the following potential function provides a final configuration that balances connectivity constraints with the requirement that the last robot approaches the target

$$f(p) = \mathbb{E}\left[\kappa_1 \sum_{i=0}^{m-1} c(p_i, p_{i+1}) + \kappa_2 \|p_m - q^*\|\right], \quad (3)$$

where $\kappa_1, \kappa_2 \in \mathbb{R}_+$ are some additional tunable parameters, and $p_0 = [0,0]^T$ is the position of the base. The expectation operator is with respect to the steady state distribution of the random variables ν_k.

If we try to compute the gradient of f, in order to implement (2), then assuming that expectation and derivative commute (this is true under weak conditions), we see that robot i needs to compute terms of the form

$$\mathbb{E}\left[\frac{\partial}{\partial p_i} c(p_i, p_{i+1})\right] = \mathbb{E}\left[\left(\log SNR_k(p_i, p_{i+1}) - \log \texttt{SNR}_{\texttt{min}}\right)\right.$$

$$\left. \times \mathbf{1}_{\{SNR_k(p_i, p_{i+1}) < \texttt{SNR}_{\texttt{min}}\}} \times \frac{\partial}{\partial p_i} h(p_i, p_{i+1})\right], \quad (4)$$

where $\mathbf{1}_{\{\cdot\}}$ is the indicator function. We also have a similar expression for $\mathbb{E}\left[\frac{\partial}{\partial p_i} c(p_{i-1}, p_i)\right]$. There are clearly major obstacles to the computation of this gradient. Most importantly, the function h and the steady-state distribution of ν_k are unknown. Even if they were known using prior measurements and simplifying models, the calculation of the resulting expectation at each period would consume significant computational resources from the robots. Yet we describe in section IV-A a simple deployment algorithm optimizing (3) which only requires that the robots have the ability to test the channel quality at each period with their neighbors, and involves no expectation computation and little coordination between the robots.
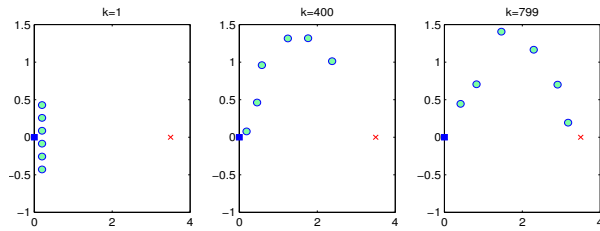

Fig. 1. Snapshots of the deployment of 6 robots with wireless connectivity constraints between a base station (blue square) and a target (red cross), following the algorithm of section IV-A. Wireless connectivity is poor except in the top part of the environment, preventing the robots to form a straight chain to the target.

B. Coverage and Vehicle Routing

Consider the coverage control problem formulated by Cortes et al. in [10]. The function to be optimized here comes from the location optimization and vector quantization literature

$$f(p) = \mathbb{E}\left[\min_{i \in \{1,\ldots,m\}} c(\|p_i - z\|)\right], \quad (5)$$

where $\|\cdot\|$ denotes the Euclidean norm, and $c : \mathbb{R}_+ \to \mathbb{R}_+$ is a continuously differentiable nondecreasing function, and $\mathbb{E}$ is the expectation with respect to the distribution of z, representing the location of a target or event of interest. This objective aims at deploying robots close to locations where targets have a high probability of appearing. The case where $c(x) = x^2$ and the target distribution has a density ϕ is considered in [10] in more details, in which case the gradient takes the form

$$\frac{\partial f}{\partial p_i}|_{p=p_k} = p_{i,k} - C_{V_{i,k}}, \quad \text{where} \quad (6)$$

$$C_{V_{i,k}} = \frac{1}{M_{V_{i,k}}} \int_{V_{i,k}} z\phi(z)dz, \quad M_{V_{i,k}} = \int_{V_{i,k}} \phi(z)dz,$$

and $V_{i,k}$ is the Voronoi cell of robot i at time k, i.e.,

$$\mathsf{V}_{i,k} = \left\{z \in \mathsf{Q} \,\middle|\, \|z - p_{i,k}\| \le \|z - p_{j,k}\|, \forall j \ne i\right\}.$$

Control law (2) then involves the computation of integrals and Voronoi cells at each step, which can require significant computational resources.

The following vehicle routing problem is closely related to the coverage control problem. At each period $k \in \mathbb{Z}_{\ge 0}$, a target appears randomly in the environment Q at position Z_k, according to the probability density ϕ. At the beginning of the period, the m robots occupy the *reference positions* $p_{1,k}, \ldots, p_{m,k}$, and the robot that can reach the target the fastest from its reference position services it. Robot i travels at speed v_i, and there are no obstacles, hence the time the kth target spends waiting for service is $\min_{i \in \{1,\ldots,m\}} \frac{1}{v_i}\|p_i - Z_k\|$. After the target is serviced, the robots can travel to new reference positions p_{k+1}. Once they have reached these new positions, a new period begins. It is not hard to see that this discrete-time problem captures the continuous time problem of [11], [23] in the limit where the arrival rate of the targets goes to 0. The goal is to minimize the steady-state waiting time of the targets

$$f(p) = \int_{\mathsf{Q}} \min_{i \in \{1,\ldots,m\}} \frac{1}{v_i}\|p_i - z\|\phi(z)dz. \quad (7)$$

Besides the potential computational difficulties involved in the calculation of the gradient of (6) or (7) however, we want in practice to deploy the robots when

the density ϕ is a priori *unknown* but one can only observe the successive positions $Z_k, k \geq 0$ of the targets. Or we may have an initial estimate of ϕ which should be refined over time based on these observations during deployment. This question was considered recently by Arsie et al. [14] for the objective (7) and Choi et al. [15], [16] for (6). In section IV-B we give a simple stochastic gradient descent algorithm optimizing the general function (5) for all such scenarios.

A Heterogeneous Coverage Problem: We can in fact significantly extend the type of coverage problems amenable to analysis, for example to heterogeneous coverage problems. Consider a vehicle routing scenario with two types of robots, m_A robots of type A and m_B robots of type B, and three types of targets a, b, ab. Targets of type a must be serviced by robots of type A, targets of type B by robots of type b, and targets of type ab by a robot of type A *and* a robot of type B. When a new target appears, it is of type α with some unknown probability λ_α, $\alpha \in \{a, b, ab\}$. The spatial distribution of targets of type α is ϕ_α and is also a priori unknown. The asymptotic configuration of the robots must now optimize the following objective, with $p = [p_1^A, \ldots, p_{m_A}^A, p_1^B, \ldots, p_{m_B}^B]$

$$f(p) = \min_p \left(\lambda_a \int_Q \frac{1}{v_A} \min_{i=1,\ldots,m_A} \|p_i^A - z\| \phi_a(z) dz \right. \quad (8)$$

$$+ \lambda_b \int_Q \frac{1}{v_B} \min_{j=1,\ldots,m_B} \|p_j^B - z\| \phi_b(z) dz + \lambda_{ab} \times$$

$$\left. \int_Q \min_{\substack{i=1,\ldots,m_A \\ j=1,\ldots,m_B}} \left\{ \max \left\{ \frac{1}{v_A} \|p_i^B - z\|, \frac{1}{v_B} \|p_j^B - z\| \right\} \right\} \phi_{ab}(z) dz \right).$$

Note that we consider a target of type ab serviced when both robots have arrived at its location. Even if all the distributions were known, computing the gradient of the objective (8) at each time step can be impractical on small platforms with limited computational power. Again a stochastic gradient algorithm optimizing (8) is quite simple to implement and works with no knowledge of the probabilities ϕ_α and λ_α, see section IV-B.

III. STOCHASTIC GRADIENT ALGORITHMS

In the previous section we argued that it would be very useful to extend the gradient descent framework for multi-robot deployment to situations where we have only access to an approximate and noisy version of the gradient (2) of the objective, or where this gradient cannot even be computed because it depends on unknown environment parameters. Stochastic gradient algorithms allow us to incorporate information gathered online during the deployment.

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

$$f(x) = \mathbb{E}_z[c(x, Z)] = \int c(x, z) d\mathbb{P}_z(z), \quad (9)$$

such as (3) or (5) for example. Here Z is a random variable modeling stochastic sources of uncertainty in the problem. The expectation cannot be computed directly if the distribution $\mathbb{P}_z$ of Z is unknown. Let us assume that c is differentiable with respect to x, for $\mathbb{P}_z$-almost all z, and denote its gradient $\nabla_x c(x, Z) := \frac{\partial c(x, z)}{\partial x}$. Finally, assume that we can observe random variables $Z_k, k \geq 0$, iid with distribution $\mathbb{P}_z$. Consider then the recursive algorithm

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

which can be rewritten in the form

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

with $h(x_k) = -\mathbb{E}_z[\nabla_x c(x_k, Z_k)|x_k]$ and $D_{k+1} = \nabla_x c(x_k, Z_k) - \mathbb{E}[\nabla_x c(x_k, Z_k)|x_k]$. Note that for Z_k a random variable, $\nabla_x c(x_x, Z_k)$ is a random vector. Define the increasing family of σ-algebras $\mathcal{F}_k := \sigma(x_0, D_i, 1 \leq i \leq k) = \sigma(x_i, 0 \leq i \leq k; D_i, 1 \leq i \leq k)$. 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 [24] says that asymptotically the sequence $\{x_k\}_{k \geq 0}$ in (11) almost surely approaches the trajectories of the ODE

$$\dot{x} = h(x). \quad (12)$$

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 engineering applications however, γ_k are chosen to converge to a small positive constant, which allows tracking of the equilibria of (12) if the problem parameters change with time. In this case, one typically obtains convergence to a neighborhood of an equilibrium of (12). The selection of proper stepsizes is an important practical issue that is not emphasized in this paper due to space constraints. It is discussed at length in references on stochastic approximation algorithms [18], [25].

Assuming now that it is valid to interchange expectation and derivation in the definition of h, we have $h(x) = -\mathbb{E}[\nabla_x c(x, Z)|x] = -\nabla f(x)$. In other words, the simple iterates (10) 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.

A. Kiefer-Wolfowitz Algorithm

Sometimes we do not even have direct access to a noisy version of the gradient of the function $f : \mathbb{R}^d \to \mathbb{R}$ to minimize, but only to noisy measurements of the function f itself. We must then reconstruct the gradient estimates, using some form of finite-difference scheme. Hence suppose that we have access to measurements of the form $\tilde{f}(x) = f(x) + \nu(x)$, where $\nu(x)$ is a random noise term with $\mathbb{E}[\nu(x)|x] = 0$. Now consider the algorithm

$$
\begin{aligned}
x_{k+1}^i &= x_k^i - \gamma_k \left(\frac{\tilde{f}(x_k + \delta e_i) - \tilde{f}(x_k - \delta e_i)}{2\delta} \right) \quad (13) \\
&= x_k^i + \gamma_k \left[-\left(\frac{f(x_k + \delta e_i) - f(x_k - \delta e_i)}{2\delta} \right) + D_{k+1}^i \right] \\
&= x_k^i + \gamma_k \left[-\frac{\partial f}{\partial x^i}(x_k) + b_k^i + D_{k+1}^i \right], \quad i = 1, \dots, d,
\end{aligned}
$$

where the zero-mean noise term D_{k+1}^i is defined by

$$
D_{k+1}^i = \frac{\nu(x_k + \delta e_i) - \nu(x_k - \delta e_i)}{2\delta},
$$

and the additional perturbation vector b_k is $O(\delta \|\nabla^2 f(x_k)\|)$, assuming that the function f is twice differentiable. Then the results of [19, chap. 5] for example imply that the iterates converge to a neighborhood of some local minimum of f. This version of the stochastic gradient algorithm using a noisy finite difference approximation of the gradient is known as the Kiefer-Wolfowitz procedure [26].

B. Spall's SPSA Algorithm

An interesting variation on the Kiefer-Wolfowitz scheme that is useful for our purpose is the *Simultaneous Perturbation Stochastic Approximation* (SPSA) of Spall [25]. In a basic version of this method we generate random variables $\Delta_k \in \mathbb{R}^d$ i.i.d., with Δ_k independent of $D_1, \dots, D_{k+1}$ and $x_0, \dots, x_k$ and $P(\Delta_k^i = 1) = P(\Delta_k^i = -1) = \frac{1}{2}$. Then we replace (13) by

$$
x_{k+1}^i = x_k^i - \gamma_k \left(\frac{\tilde{f}(x_k + \delta \Delta_n) - \tilde{f}(p_k)}{\delta \Delta_k^i} \right), \quad (14)
$$

where $\tilde{f}(x) = f(x) + \nu(x)$. Again the iterates generally converge to a neighborhood of a minimum of f almost surely [19], [25]. Note that for $f : \mathbb{R}^d \to \mathbb{R}$, (14) requires only 2 function evaluations instead of $2d$ for (13) !

IV. ADAPTIVE DEPLOYMENT ALGORITHMS

We now revisit the examples of section II and discuss the application of stochastic gradient algorithms in these scenarios.

A. Wireless Deployment with Random Fading Channels

Consider the deployment problem with wireless connectivity constraints of section II-A. What is required to implement a stochastic gradient descent algorithm for deployment is an estimate of the expectation (4). Two successive robots i and $i+1$ in the chain can test the quality of the channel connecting them at each period. At period k, they measure the random value $m_k^{i,i+1} := \log SNR_k(p_{i,k}, p_{i+1,k})$. The quantity (4) also involves the computation of $\frac{\partial}{\partial p_i} h(p_i, p_{i+1})$, for which we construct a finite difference estimate using the SPSA algorithm. More precisely, at period k, after the robots obtained the quantity $m_k^{i,i+1}$, they all take random steps as follows. For $i = 1, \dots, m$, robot i generates random variables $\Delta_{i,k} = [\Delta_{i,k}^1, \Delta_{i,k}^2]^T$ as in the previous paragraph, and moves to $p_{i,k} + \delta\Delta_{i,k}$, with δ sufficiently small. Again, the robots test the channel quality with their neighbors in the chain, so that robot i collects the value $\hat{m}_k^{i,i+1} := \log SNR_k(p_{i,k} + \delta\Delta_{i,k}, p_{i+1,k} + \delta\Delta_{i+1,k})$. Now consider the following Taylor expansion, with $p_i = [p_i^1, p_i^2]^T$

$$
\begin{aligned}
&\frac{h(p_{i,k} + \delta\Delta_{i,k}, p_{i+1} + \delta\Delta_{i+1,k}) - h(p_{i,k}, p_{i+1,k})}{\delta\Delta_{i,k}^1} \\
&\approx \frac{\partial h}{\partial p_i^1}(p_{i,k}, p_{i+1,k}) + \frac{\partial h}{\partial p_i^2}(p_{i,k}, p_{i+1,k}) \frac{\Delta_{i,k}^2}{\Delta_{i,k}^1} \\
&+ \sum_{j=1}^2 \frac{\partial h}{\partial p_{i+1}^j}(p_{i,k}, p_{i+1,k}) \frac{\Delta_{i+1,k}^j}{\Delta_{i,k}^1}.
\end{aligned}
$$

All the terms except the first one have zero mean hence enter as additional noise terms in the stochastic approximation. In other words, the quantity $(\hat{m}_k^{i,i+1} - m_k^{i,i+1})/(\delta\Delta_{i,k}^1)$ is, up to second order terms, an unbiased estimate of $\partial h/\partial p_i^1$. We can reason similarly for the other partial derivatives $\partial h/\partial p_i^2$ required by robot i to perform its gradient descent. In summary, a stochastic approximation of the expression (4) is obtained at period k by

$$
(m_k^{i,i+1} - \log \mathtt{SNR}_{\min}) \begin{bmatrix} \frac{\hat{m}_k^{i,i+1} - m_k^{i,i+1}}{\delta\Delta_{i,k}^1} \\ \frac{\hat{m}_k^{i,i+1} - m_k^{i,i+1}}{\delta\Delta_{i,k}^2} \end{bmatrix} \times
$$

$$
\mathbf{1}_{\{SNR_k(p_i, p_{i+1}) < \mathtt{SNR}_{\min}\}}.
$$

This expression, which depends only on quantities that robot i can obtain by direct interaction with its neighbors

in the chain, is then used in place of (4) in the gradient descent. The almost sure convergence to a neighborhood of a local minimum of f defined by (3) follows directly from the analysis of the Kiefer-Wolfowitz or SPSA procedure. A small deployment example was presented on Fig. 1.

B. Adaptive Coverage and Vehicle Routing

Consider the coverage problems of section II-B, where the distribution ϕ of the targets is now unknown. At each period, a target is present at position $Z_k \in Q$, and we assume that at least the robot closest to the target can observe it. The successive positions Z_k could also correspond for example to a single target with Markovian dynamics, as long as a stationary distribution ϕ exists. At the end of each period, the robots can change their reference points in order to optimize the steady-state objective (5). This problem fits the expectation minimization framework discussed in section III. In particular, the stochastic gradient descent laws to implement are typically much easier to compute than the corresponding deterministic gradient updates. Indeed, equation (10) simplifies to

$$p_{i,k+1} = \begin{cases} p_{i,k} + \gamma_k c'(\|z_k - p_{i,k}\|) \frac{z_k - p_{i,k}}{\|z_k - p_{i,k}\|} \\ \qquad \text{if } i \text{ is closest to } z_k, \\ p_{i,k} \qquad \text{otherwise.} \end{cases} \quad (15)$$

Note that no Voronoi cell computation or integration is required, only a distributed mechanism to find which robot is the closest to the target (e.g. a simple min-consensus scheme). Only the closest robot updates its reference position for the period. We can then specialize (15) to the standard coverage case with $c(x) = x^2$, which gives the update $p_{i,k} + \gamma_k(z_k - p_{i,k})$ for the closest robot. This particular adaptive algorithm has been used extensively in various fields [27]–[29] and its convergence as a stochastic gradient algorithm analyzed in [30], [31].

For the vehicle routing objective (7), we obtain the update $p_{i,k} + \gamma_k \frac{z_k - p_{i,k}}{\|z_k - p_{i,k}\|}$ for the closest robot, which is somewhat different from the quadratic case because the stepsize does not vanish as the distance to the target becomes small (a projection of the steps on Q can prevent the iterates to leave Q). With the corresponding update law, the robots converge to the so-called median Voronoi configuration, which is a local minimum of (7)[1]. This asymptotic configuration is also obtained by [14], by a more complicated update law (there a robot should

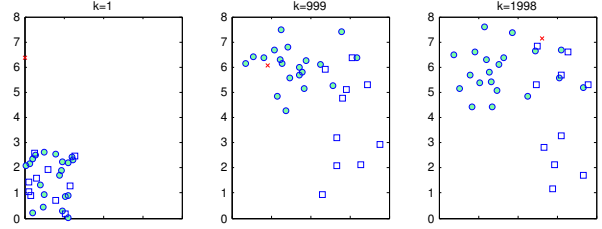[1]a rigorous convergence proof will be presented in the full version of this paper.

Fig. 2. Vehicle routing for a system with two types of vehicles, A (full circles) and B (empty squares). Only the reference points of the vehicles at the beginning of the periods are shown. Targets requiring service from type A appear with probability 40% and a distribution centered at $[2; 6]^T$. Targets of type B appear with probability 20% and a distribution centered at $[6; 2]^T$. Finally targets of type AB appear with probability 40% and a distribution centered at $[6; 6]^T$. Note how vehicles of type A and B tend to pair in order to service the targets of type AB efficiently (here $v_A = v_B$).

compute the median of all the targets it visited in the past every times it moves).

Heterogeneous Coverage: In addition to simplifying the convergence proofs, the stochastic gradient point of view allows us to find simple update laws for more complex problems. To illustrate this point, consider the routing problem with heterogeneous vehicles discussed at the end of subsection II-B. One can verify that the stochastic gradient update rule takes the following form. When a target of type a appears, the closest robot of type A moves toward it by a step $\gamma_k \frac{z_k - p_{i,k}}{\|z_k - p_{i,k}\|}$, and similarly for a target of type b. If the target is of type ab, the closest A and B robots first find which of the two is the *farthest* from the target. Then only this robot moves by the step $\gamma_k \frac{z_k - p_{i,k}}{\|z_k - p_{i,k}\|}$. 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.

V. CONCLUSION

We propose a general framework for a range of robotic network deployment scenarios where information about stochastic sources of uncertainty needs to be collected online to optimize the deployment. This paper gives an overview of the framework, which is based on stochastic gradient descent algorithms and the related theory of stochastic approximations. The algorithms are very flexible in the type of uncertainties they can handle.

Among their known drawbacks, stochastic gradient algorithms can be slow compared to their deterministic counterparts, and their behavior is sensitive to the choice of the stepsizes γ_k. In practice, one could obtain a first deployment configuration using a deterministic deployment algorithm based on prior information about the environment, followed by the stochastic gradient

scheme accounting for the updated information collected by the robots and the environment modeling errors. The simplicity of the stochastic deployment algorithms makes them ideal candidates for implementation on small platforms with limited computational power. There are also many other deployment problems not discussed here that can benefit from this approach. For example, the convergence proof of various source seeking schemes, such as the one considered in [1], can be simplified and shown to hold under general assumptions on the noise characteristics, by viewing it essentially as a Kiefer-Wolfowitz procedure. Formation control using noisy observations and communication links can also be studied from this point of view: the recent papers [32]–[34] study the related stochastic consensus problem using stochastic approximations.

REFERENCES

[1] 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.

[2] 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.

[3] G. Hollinger, A. Kehagias, and S. Singh, "Efficient, guaranteed search with multi-agent teams," in *Robotics: Science and Systems*, July 2009.

[4] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.

[5] T. Balch and R. Arkin, "Behavior-based formation control for multirobot teams," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 6, pp. 926–939, 1998.

[6] J. Reif and H. Wang, "Social potential fields: A distributed behavioral control for autonomous robots," *Robotics and Autonomous Systems*, vol. 27, pp. 171–194, 1999.

[7] F. Schneider and D. Wildermuth, "A potential field based approach to multi-robot formation navigation," in *Proceedings of International Conference on Robotics, Intelligent Systems and Signal Processing*, 2003, pp. 680–685.

[8] V. Gazi and K. Passino, "A class of attractions/repulsion functions for stable swarm aggregations," *International Journal of Control*, vol. 77, no. 18, pp. 1567–1579, 2004.

[9] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, March 2006.

[10] J. Cortés, S. Martinez, 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.

[11] 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.

[12] K. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE Control Systems Magazine*, vol. 22, no. 3, pp. 52–67, 2002.

[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, 2009.

[14] A. Arsie, K. Savla, and E. Frazzoli, "Efficient routing algorithms for multiple vehicles with no explicit communications," *IEEE Transactions on Automatic Control*, 2009, in Press.

[15] J. Choi, S. Oh, and R. Horowitz, "Distributed learning and cooperative control for multi-agent systems," *Automatica*, vol. 45, no. 12, pp. 2802–2814, December 2009.

[16] 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.

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

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

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

[20] O. Burdakov, P. Doherty, K. Holmberg, J. Kvarnström, and P.-M. Olsson, "Positioning unmanned aerial vehicles as communication relays for surveillance tasks," in *Proceedings of Robotics: Science and Systems*, Seattle, USA, June 2009.

[21] A. Goldsmith, *Wireless Communications*. Cambridge University Press, 2005.

[22] A. Ghaffarkhah and Y. Mostofi, "Communication-aware navigation functions for robotic networks," in *Proceedings of the American Control Conference*, 2009.

[23] D. Bertsimas and G. V. Ryzin, "A stochastic and dynamic vehicle routing problem in the Euclidean plane," *Operations Research*, vol. 39, pp. 601–615, 1991.

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

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

[26] J. Kiefer and J. Wolfowitz, "Stochastic estimation of the maximum of a regression function," *Annals of Mathematical Statistics*, vol. 23, no. 3, pp. 462–466, September 1952.

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

[28] 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.

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

[30] L. Bottou, "Stochastic gradient learning in neural networks," in *Proceedings of Neuro-Nimes 91*, Nimes, France, 1991.

[31] G. Pages, "A space quantization method for numerical integration," *Journal of Computational and Applied Mathematics*, vol. 89, pp. 1–38, 1997.

[32] R. Rajagopal and M. Wainwright, "Network-based consensus averaging with general noisy channels," in *Proceedings of the Allerton Conference on Communication, Control, and Computing*, 2008.

[33] S.Kar and J. Moura, "Distributed consensus algorithms in sensor networks with imperfect communication: Link failures and channel noise," *IEEE Transactions on Signal Processing*, vol. 57, no. 1, pp. 355–369, January 2009.

[34] M. Huang and J. Manton, "Stochastic consensus seeking with noisy and directed inter-agent communication: Fixed and randomly varying topologies," *IEEE Transactions on Automatic Control*, vol. 55, no. 1, pp. 235–241, January 2010.