3-25-2010

Adaptive Robot Deployment Algorithms

Jerome LE NY
University of Pennsylvania, jerome.le-ny@polymtl.ca

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

Adaptive Robot Deployment Algorithms

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 static deployment 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 deployment algorithms that may not be adaptive to uncertain dynamic environments. We address this challenge by showing that algorithms for a variety of deployment scenarios in 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 adapt to slow temporal variations in environmental parameters. To illustrate the richness of the framework, we discuss several applications, including searching for a field extrema, deployment with stochastic connectivity constraints, coverage, and vehicle routing scenarios.

Keywords
robotics, potential field methods, stochastic gradient descent algorithms, stochastic approximation

Disciplines
Controls and Control Theory | Robotics

Adaptive Robot Deployment Algorithms

Jerome Le Ny, George J. Pappas
School of Electrical and Systems Engineering
University of Pennsylvania, Philadelphia, PA 19104
Email: jleny, pappasg@seas.upenn.edu

***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 static deployment 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 deployment algorithms that may not be adaptive to uncertain dynamic environments. We address this challenge by showing that algorithms for a variety of deployment scenarios in 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 adapt to slow temporal variations in environmental parameters. To illustrate the richness of the framework, we discuss several applications, including searching for a field extrema, deployment with stochastic connectivity constraints, coverage, and vehicle routing scenarios.**

I. Introduction

There has been in the last few years a significant research effort 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 [27, 19, 18]. 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 [21], 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 [3, 31, 33, 14, 28], coverage [19, 11] as well as certain vehicle routing problems [13], or foraging and source seeking [29, 27]. Most of this work assumes a deterministic 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 [27, 34, 2, 10, 9]. 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. A number of classical and new examples are also introduced, including source seeking, deployment with wireless connectivity constraints, and coverage scenarios, and we point out some deficiencies of deterministic models. Section III recalls the powerful framework of stochastic approximations and the special case of stochastic gradient descent algorithms. Finally, in section IV we revisit the scenarios of section II and illustrate how stochastic gradient descent 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.

II. Deficiencies of Static Deployment Algorithms

In the basic form of 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 [21, 11]. 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. $p_{i,k} \in \mathsf{Q}$, for $i = 1, \ldots, m$. Here we assume a discrete-time model for simplicity and direct correspondence with optimization algorithms, but a continuous-time model could be used as well. At the high-level planning stage we assume that we can work

with a fully actuated model

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

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 well-designed objective function $f : \mathbb{Q}^m \to \mathbb{R}$, 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

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

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

Finally, we briefly comment on the fact that potential function methods can get trapped in local minima of the potential. The stochastic gradient descent algorithms described later do not avoid this issue, however we point out that they tend to escape *shallow* local minima due to the intrinsic problem noise, and moreover can be easily combined with simulated annealing type algorithms by injecting additional artificial noise in order to asymptotically reach a global minimum of the potential [36]. However, simulated annealing algorithms can be slow or require delicate tuning.

A. Source Seeking

In source seeking problems, one or several robots try to reach the minimum of a scalar field $T : \mathbb{Q} \to \mathbb{R}$, which directly represents a physical quantity of interest, such as the concentration of some chemical in the environment, see e.g. [27]. Assume for now that the robots adopt an (arbitrary) rigid formation, with center of mass M_k at time k, which can also be enforced using potential field methods [31, 27]. Then this formation can move in order for M_k to descend the gradient of T, in other words we implement (2) in terms of M_k instead of p_k. Hence the desired dynamics of the center of mass take the form

$$M_{k+1} = M_k - \gamma_k \nabla T(M_k). \qquad (3)$$

In practice however, the robots can measure T but not its gradient, which must then be approximated by some form of finite-difference scheme. A further complication arises due to the fact that the measurements are subject to noise.

We propose the following algorithm, assuming that the robots can share their field measurements at their respective positions in order to estimate $\nabla T(M_k)$, can maintain the rigid formation at all times and can stay relatively close to M_k in order for the finite-difference approximation error to remain small. At period k the robot positions are $p_k = [p_{1,k}, \dots, p_{m,k}]$ and these robots take measurements $\{y_{1,k}, \dots, y_{m,k}\}$ of the field, of the form

$$y_{i,k} = T(p_{i,k}) + \nu_{i,k}, \; i = 1, \dots, m, \qquad (4)$$

where $\nu_{i,k}$ a measurement and possibly environmental noise. For simplicity we consider a two dimensional environment, but extension to three dimensions is immediate. Denote

$$p_{i,k} = M_k + \Lambda_{i,k}, \text{ with } M_k = \frac{1}{m}\sum_{k=1}^{m} p_{i,k}.$$

Next, consider a finite difference approximation

$$\frac{\partial T}{\partial x}(M_k) \approx \sum_{i=1}^{m} w_{i,x} T(p_{i,k}), \;\; \frac{\partial T}{\partial y}(M_k) \approx \sum_{i=1}^{m} w_{i,y} T(p_{i,k}).$$

Well-known techniques are available to obtain good weights $w_{x,i}$ and $w_{y,i}$ providing a high-order finite difference approximation, for example based on Lagrange interpolation or the method of undetermined coefficients [1]. The standard central difference scheme would correspond to four robots forming a square, but the weights can be computed for an arbitrary number of robots forming an arbitrary shape. For example, for a formation with 3 robots forming an equilateral triangle $\Lambda_{i,k} = \delta \begin{bmatrix} \cos\left(\frac{2i\pi}{3}\right) \\ \sin\left(\frac{2i\pi}{3}\right) \end{bmatrix}$ with δ sufficiently small, we obtain by the method of undetermined coefficients

$$\begin{bmatrix} w_{1,x} \\ w_{2,x} \\ w_{3,x} \end{bmatrix} = \frac{1}{\delta} \begin{bmatrix} 2/3 \\ -1/3 \\ -1/3 \end{bmatrix}, \quad \begin{bmatrix} w_{1,y} \\ w_{2,y} \\ w_{3,y} \end{bmatrix} = \frac{1}{\delta} \begin{bmatrix} 0 \\ \sqrt{3}/3 \\ -\sqrt{3}/3 \end{bmatrix}.$$

Then we impose that the formation moves according to

$$M_{k+1} = M_k - \gamma_k W_k Y_k, \qquad (5)$$

where

$$W_k = \begin{bmatrix} w_{1,x} & \cdots & w_{m,x} \\ w_{1,y} & \cdots & w_{m,y} \end{bmatrix} \text{ and } Y_k = \begin{bmatrix} y_{1,k}, \cdots, y_{m,k} \end{bmatrix}^T.$$

This problem is a typical example where the gradient descent (2) can only be implemented approximately, in the form of (5), with the additional complication of the noise making the iterates M_k random. The convergence of similar source seeking schemes, e.g. the one presented in [27], is often studied in a deterministic framework by assuming that the noise can be neglected, which is hard to justify. In section IV-A, we show that M_k following (5) with appropriately chosen stepsizes γ_k converges to a neighborhood of a local minimum of T almost surely, under very weak conditions on the noise,

e.g. that $\mathbb{E}[\nu_k|p_k] \to 0$ as $k \to \infty$ and that the random variables ν_k have a finite second moment. As an additional benefit of the stochastic approximation point of view, we describe later a simple scheme allowing a *single* robot to descent the gradient of T with performance comparable to that of the robot formation, which is clearly useful when maintaining such a formation is difficult.

B. 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 $\mathsf{Q} \subset \mathbb{R}^2$, 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. [8] 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 [16]. 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 [15] 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. To our knowledge, this paper present the first approach that can adapt to an unknown SNR map, as described in section IV-B.

We assume that the random variables ν_k have a steady-state distribution as $k \to \infty$, denote a generic random variable with this distribution by ν_∞, and the corresponding generic random SNR value by SNR_∞, so that $\log SNR_\infty(p_1,p_2) = h(p_1,p_2)+\nu_\infty$. For simplicity, we assume that the distribution of ν_∞ is independent of p_1, p_2, although more refined models could be considered as well. 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 $\mathtt{SNR}_{\mathtt{min}}$

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

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 approach the target

$$f(p) = \mathbb{E}\left[\kappa_1 \sum_{i=0}^{m-1} c(p_i,p_{i+1}) + \kappa_2 \|p_m - q^*\|\right], \qquad (6)$$

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

$$\mathbb{E}\left[\frac{\partial}{\partial p_i} c(p_i,p_{i+1})\right] = \mathbb{E}\Bigg[\Big(\log SNR_\infty(p_i,p_{i+1}) - \log \mathtt{SNR}_{\mathtt{min}}\Big) \times \mathbf{1}_{\{SNR_\infty(p_i,p_{i+1}) < \mathtt{SNR}_{\mathtt{min}}\}} \times \frac{\partial}{\partial p_i} h(p_i,p_{i+1})\Bigg], \qquad (7)$$

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 distribution of ν_∞ 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-B a simple deployment algorithm optimizing (6) 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.

C. Coverage and Vehicle Routing

Consider the coverage problem formulated by Cortes et al. in [11]. The function to be optimized here comes from the location optimization and vector quantization literature

$$\begin{aligned} f(p) &= \int_{\mathsf{Q}} \min_{i\in\{1,\ldots,m\}} c(d_i(p_i,z))\phi(z)dz \\ &= \mathbb{E}\left[\min_{i\in\{1,\ldots,m\}} c(d_i(p_i,z))\right], \end{aligned} \qquad (8)$$

where $d_i : \mathsf{Q}\times\mathsf{Q} \to \mathbb{R}_+$ is a distance function, $c : \mathbb{R}_+ \to \mathbb{R}_+$ is a nondecreasing function, and $\phi : \mathsf{Q} \to \mathbb{R}_+$ is a known probability density function. Intuitively, $\phi(z)$ represents some

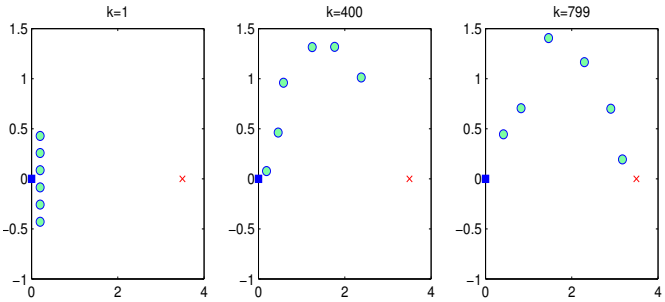

Fig. 1. Snapshots of the deployment of 6 robots with wireless connectivity constraints between a base station (blue square) and a target (red cross). Wireless connectivity is poor except in the top part of the environment, preventing the robots to form a straight chain to the target.

prior knowledge about the importance of deploying a robot close to position z. The specific case where $d_i(p_i, z) = \|p_i - z\|$ is the Euclidean distance and $c(x) = x^2$ is considered in [11] in more details, in which case the gradient takes the form

$$\frac{\partial f}{\partial p_i}|_{p=p_k} = p_{i,k} - C_{V_{i,k}}, \text{ where} \tag{9}$$
$$C_{V_{i,k}} = \frac{1}{M_{\mathsf{V}_{i,k}}} \int_{\mathsf{V}_{i,k}} z\phi(z)dz, \quad M_{\mathsf{V}_{i,k}} = \int_{\mathsf{V}_{i,k}} \phi(z)dz,$$

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

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

Control law (2) then involves the computation of integrals and Voronoi cells at each step, and the problem becomes more complicated if other functions c and distances are considered.

The following vehicle routing problem is closely related to the previous coverage problem. At each period $k \in \mathbb{Z}_{\geq 0}$, 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 k^{th} target spends waiting for service is $\min_{i\in\{1,\ldots,m\}} \frac{1}{v_i}\|p_i - Z_k\|$ (with obstacles, one should replace the Euclidean distance by the shortest distance to the target). 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 can be used to analyze the continuous time problem of [5, 13] 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

$$f(p) = \int_{\mathsf{Q}} \min_{i\in\{1,\ldots,m\}} \frac{1}{v_i}\|p_i - z\|\phi(z)dz, \tag{10}$$

which is nothing but the coverage problem with $c(x) = x$ and $d_i(p_i, z) = \|p_i - z\|/v_i$. Finding a configuration achieving a global minimum of (10) is known to be an NP-hard problem [12], but we can still implement the gradient descent controller (2) to asymptotically reach reference positions which form a local minimum of f.

Besides the potential computational difficulties involved in the calculation of the gradient of (9) or (10) 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. [2] for the objective (10) and Choi et al. [10, 9] for (9). In section IV-C we give a new simple stochastic gradient descent algorithm optimizing the general function (8) for all such scenarios.

A Heterogeneous Coverage Problem: We can in fact significantly extend the complexity of the type of coverage problems amenable to analysis, for example to heterogeneous coverage problem. For example, 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 $p = [p_1^A, \ldots, p_{m_A}^A, p_1^B, \ldots, p_{m_B}^B]$

$$f(p) = \min_p \left(\lambda_a \int_{\mathsf{Q}} \frac{1}{v_A} \min_{i=1,\ldots,m_A} \|p_i^A - z\|\phi_a(z)dz \right. \tag{11}$$
$$+ \lambda_b \int_{\mathsf{Q}} \frac{1}{v_B} \min_{j=1,\ldots,m_B} \|p_j^B - z\|\phi_b(z)dz + \lambda_{ab} \times$$
$$\left. \int_{\mathsf{Q}} \min_{\substack{i=1,\ldots,m_A \\ j=1,\ldots,m_B}} \left\{ \max\left\{ \frac{1}{v_A}\|p_i^B - z\|, \frac{1}{v_B}\|p_j^B - z\| \right\} \right\} \phi_{ab}(z)dz \right).$$

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 (11) at each time step can be impractical on small platforms with limited computational power. Again a stochastic gradient algorithm optimizing (11) and described in section IV-C is quite simple to implement and works with no knowledge of the probabilities ϕ_α and λ_α.

III. Stochastic Approximations

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. We will see in the next section that the this can be done using stochastic versions of the gradient descent scheme, and moreover, computing the approximate gradient descent directions is often simpler than computing (2). The convergence analysis of these algorithms relies on the theory of stochastic approximations [25, 4, 24], and we present a few important ideas of this theory here.

For the purpose of this paper, it is usually sufficient to consider stochastic approximation (SA) algorithms of the form

$$x_{k+1} = x_k + \gamma_k(h(x_k) + b_k + D_{k+1}), \tag{12}$$

where the sequence $\{x_k\}_k$ consists of some random iterates whose asymptotic behavior is of interest, D_k is some random zero-mean noise, b_k is a small bias term that can be asymptotically neglected, and γ_k is a small step-size. More precise assumptions about these different terms appear below. The main idea however is that under broad conditions the trajectories of the iterates x_k verifying (12) follow asymptotically the integral curves of the Ordinary Differential Equation (ODE)

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

Note that if $h(x) = -\nabla f(x)$ for some real-valued function f, this "ODE method" [25] says that the behavior of the noisy iterates (12) can still be analyzed via the simple deterministic gradient flow $\dot{x} = -\nabla f(x(t))$, just as for the deterministic gradient descent algorithm (2). Let $\mathcal{S} = \{x : \nabla f(x) = 0\}$ denote the set of equilibrium points of the ODE, which is also the set of critical points of f. Stochastic approximation theorems give broad conditions under which the iterates $\{x_k\}$ converge to $\mathcal{S}$. In general, h is assumed Lipshitz, in order to guarantee existence and uniqueness of solutions to the ODE (13). However, the theory extends to differential inclusions [24], [6, chapter 5], which allows us to consider functions f that are not differentiable everywhere and replace gradients by subgradients. This situation arises in the vehicle routing problem since the function $p \to \|p-z\|$ is not differentiable at z. The following set of additional assumptions, by no means the weakest possible, allow us to obtain useful convergence theorems [6]

(A1) The sequence $\{b_k\}$ is bounded and $b_k \to 0$ as $k \to \infty$ almost surely (a.s.).

(A2) The stepsizes $\{\gamma_k\}$ are positive bounded scalars satisfying $\sum_k \gamma_k = \infty$, $\sum_k \gamma_k^2 < \infty$ a.s.

(A3) We have $E[D_{k+1}|\mathcal{I}_k] = 0$, where $\mathcal{I}_k = \{x_m, \gamma_m, b_m, D_m, m \leq k\}$. Furthermore, the random variables $\{D_k\}$ are square integrable, with

$$E[\|D_{k+1}\|^2|\mathcal{I}_k] \leq K(1 + \|x_k\|^2) \text{ a.s.}, k \geq 0,$$

for some constant $K > 0$.

(A4) The iterates (12) remain bounded a.s., i.e.,

$$\sup_k \|x_k\| < \infty, \text{ a.s.}$$

Assumption (A4) is the most challenging to verify in the general theory of SA algorithms, but in our case it is typically dealt with by assuming that the environment Q of the robots is bounded. Under the previous assumptions, if $\mathcal{S}$ is a discrete set, then $\{x_k\}_k$ converges a.s. to a point of $\mathcal{S}$, possibly sample path dependent [6]. Moreover under typical conditions the convergence point is a local minimum of f (rather than a saddle point or maximum).

A. Expectation Minimization Problems

A common case where we have access to a noisy version of the gradient of the function f to minimize arises when we assume that f is of the form

$$f(x) = \mathbb{E}[c(x, Z)] = \int c(x, z) d\mathbb{P}(z), \tag{14}$$

where Z is a random variable and the expectation operator is with respect to its distribution (which is assumed here to be independent of x). Often the expectation is either difficult or impossible to compute directly, due to the complexity of the function c or because the distribution of Z is hard to evaluate or simply unknown. The function values $c(x, Z)$ for given inputs x, Z can be observed, however.

Let us assume that $(x, z) \mapsto c(x, z)$ is differentiable with respect to x, for almost all z, and that we know $\nabla_x c(x, z)$. Note that for Z a random variable, $\nabla_x c(x, Z)$ is a random vector, called the stochastic gradient of c. Consider then the SA algorithm

$$x_{k+1} = x_k - \gamma_k \nabla_x c(x_k, Z_k), \tag{15}$$

which can be rewritten in the standard form (12)

$$x_{k+1} = x_k - \gamma_k(\mathbb{E}[\nabla_x c(x_k, Z_k)|x_k] + D_{k+1})$$

where $D_{k+1} = \nabla_x c(x_k, Z_k) - \mathbb{E}[\nabla_x c(x_k, Z_k)|x_k]$ satisfies (A3). Suppose now that it is valid to interchange expectation and derivation in the previous equation, so that

$$\begin{aligned}\mathbb{E}[\nabla_x c(x, Z)|x] &= \int \nabla_x c(x, z) d\mathbb{P}(z) \\ &= \nabla \int c(x, z) d\mathbb{P}(z) = \nabla f(x).\end{aligned} \tag{16}$$

Here we have used the fact that the distribution of Z is independent of x. Hence in this case $\nabla_x c(x, Z)$ is an unbiased estimate of the gradient of f at x. We see then that the ODE associated to (15) is the gradient flow

$$\dot{x} = -\nabla f(x(t)).$$

In particular under broad conditions such as the ones presented previously, the iterates of (15) converge to a local minimum of f almost surely. Conditions guaranteeing the interchange of integral and derivation in (16) can be obtained from the dominated convergence theorem [32] and are usually satisfied. In fact, this interchange could even be valid only approximately up to a small bias term b_k that converges to 0 asymptotically as in assumption (A1).

B. 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

$$x_{k+1}^i = x_k^i - \gamma_k \left(\frac{\tilde{f}(x_k + \delta e_i) - \tilde{f}(x_k - \delta e_i)}{2\delta} \right) \quad (17)$$
$$= x_k^i + \gamma_k \left[- \left(\frac{f(x_k + \delta e_i) - f(x_k - \delta e_i)}{2\delta} \right) + D_{k+1}^i \right]$$
$$= x_k^i + \gamma_k \left[- \frac{\partial f}{\partial x^i}(x_k) + b_k^i + D_{k+1}^i \right], \ i = 1, \ldots, d,$$

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. Here this term does not satisfy assumption (A1), however if it remains small then the results of [6, chap. 5] for example imply that the iterates converge to a neighborhood of some local minimum of f. This version of the stochastic gradient descent algorithm using a noisy finite difference approximation of the gradient is known as the Kiefer-Wolfowitz procedure [22].

C. 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 [36]. 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 $D_1, \ldots, D_{k+1}$ and $x_0, \ldots, x_k$ and $P(\Delta_k^i = 1) = P(\Delta_k^i = -1) = \frac{1}{2}$. Then we replace (17) by

$$x_{k+1}^i = x_k^i - \gamma_k \left(\frac{\tilde{f}(x_k + \delta \Delta_n) - \tilde{f}(p_k)}{\delta \Delta_k^i} \right), \quad (18)$$

where $\tilde{f}(x) = f(x) + \nu(x)$. Again the iterates converge to a neighborhood of a minimum of f almost surely [36, 6]. Note that for $f : \mathbb{R}^d \to \mathbb{R}$, (18) requires only 2 function evaluations instead of $2d$ for (17) !

D. Choice of Stepsizes and Tracking

One potential issue associated with stochastic gradient algorithms is that their practical performance is sensitive to the tuning of the stepsizes γ_k. The choice of these stepsizes has been extensively investigated [24, 36]. Assumption (A2), which is satisfied for example by $\gamma_k = 1/(k+1)$, is a typical assumption in theorems that prove asymptotic convergence to a fixed point. In practical applications, other choices are possible, in particular letting γ_k converge asymptotically to a small constant. In that case, we usually only have convergence to a neighborhood of an equilibrium of the ODE. However, asymptotically constant stepsizes allow us to *track* slowly varying changes in the system [4]. In our robotic applications this means that the configuration can adapt to slow variations in the environment parameters.

IV. Adaptive Deployment Algorithms

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

A. Source Seeking with Noisy Measurements

Recall that in subsection II-A, we wanted to understand the asymptotic behavior of the center of mass of the rigid formation following the update (5), in the case where the robots take the noisy measurements (4). We can now see that this scheme is essentially the Kiefer-Wolfowitz procedure of section III-B (with a more general finite difference approximation), hence we know that M_k will converge to a neighborhood of a local minimum of T almost surely. In contrast to [27], the argument does not rely on the possibility of neglecting the noise, which can be quite large, or on any assumption on the noise distribution except for the fact that its mean converges asymptotically to zero. The distribution of $\nu_{i,k}$ can even depend on the position $p_{i,k}$.

In addition, a single robot taking noisy measurements can descend along the gradient of T using the following algorithm derived from Spall's SPSA algorithm, see subsection III-C. We divide period k into two steps. First, the robot takes a noisy measurement $y_k = T(p_k) + \nu_k$ at its current position, and generates a random vector $\Delta_k = [\Delta_k^1, \Delta_k^2]^T$, with $\mathbb{P}(\Delta_k^i = -1) = \mathbb{P}(\Delta_k^i = +1) = \frac{1}{2}$. In the first phase of the period, it moves by a small amount $\delta \Delta_k$ to get the noisy measurement $\hat{y}_k = T(p_k + \delta \Delta_k) + \hat{\nu}_k$. In the second phase of the period, the agent moves to the point $p_{k+1} = [p_{k+1}^1, p_{k+1}^2]^T$ with

$$p_{k+1}^j = p_k^j - \gamma_k \frac{\hat{y}_k - y_k}{\delta \Delta_k^j}, \ j = 1, 2.$$

Note that in general, using more robots moving in formation and averaging their measurements clearly improves the quality of the gradient estimates, by simply averaging the noise faster than in the SPSA algorithm. However, the exact relationship between the quality of the gradient approximations and the speed of convergence of the stochastic gradient descent algorithm is not straightforward to establish. In fact, there is some empirical evidence suggesting that the SPSA algorithm can perform as well as the Kiefer-Wolfowitz procedure in practice [36], even though it operates with much less measurements. This would imply that a single robot on average can converge as fast as a robot formation to a minimum of T.

B. Wireless Deployment with Random Fading Channels

Consider the deployment problem with wireless connectivity constraints of section II-B. What is required to implement a stochastic gradient descent algorithm for deployment is an estimate of the expectation (7). 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 (7) 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 $i = 1, \ldots, m$, 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 (7) is obtained at period k by

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

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 (7) in the gradient descent. The almost sure convergence to a neighborhood of a local minimum of f defined by (6) follows directly from the existing analysis of the Kiefer-Wolfowitz or SPSA procedure. A small deployment example was presented on Fig. 1.

C. Adaptive Coverage and Vehicle Routing

Consider the coverage problems of section II-C, where the distribution ϕ of the targets is now unknown. At each period, a target is present at position $Z_k \in \mathsf{Q}$, 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 (8). We now see that this problem fits the expectation minimization framework discussed in section III-A. In particular, the stochastic gradient descent laws to implement are typically much easier to compute than the corresponding deterministic gradient updates. Indeed, equation (15) simplifies to

$$p_{i,k+1} = \begin{cases} p_{i,k} - \gamma_k \frac{\partial c(d_i(z_k - p_{i,k}))}{\partial p_i} & \text{if } i \text{ is closest to } z_k, \\ p_{i,k} & \text{otherwise.} \end{cases} \tag{19}$$

Note that no Voronoi cell computation or integration is required, only a distributed mechanism to find which robot is the closest to the target. Only the closest robot updates its reference position for the period. We can then specialize (19) 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. It is an adaptive vector quantization algorithm also known as LBG [17], the K-means algorithm of MacQueen [26], and it is related to Kohonen's self-organizing maps [23]. The fact that this algorithm is a stochastic gradient descent algorithm is discussed by Bottou [7]. The recent papers [9, 10] have also considered certain cases of the coverage problem with quadratic function c from the point of view of stochastic approximations.

For the vehicle routing objective (10), 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. With the corresponding update law, the robots converge to the so-called median Voronoi configuration, which is a local minimum of (10). This asymptotic configuration is also obtained by [2], by a somewhat more complicated update law (there a robot should compute the median of all the targets it visited in the past every times it moves). Consequently, their convergence proof is quite complex, whereas we see that for our update law the convergence is an immediate consequence of the SA theory.

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-C. One can immediately 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

This paper proposes a general framework for a range of robotic network deployment scenarios, based on stochastic gradient descent algorithms and the related theory of stochastic approximations. The framework is very flexible in the type of uncertainties it can handle. Among their known drawbacks, stochastic gradient descent algorithms can be slow compared to their deterministic counterparts, and the asymptotic behavior of the algorithms is sensitive to the choice of the stepsizes γ_k. Keeping γ_k asymptotically constant allows us to track slow variations in the environment parameters [4]. Regarding the issue of convergence speed, we could in deployment problems

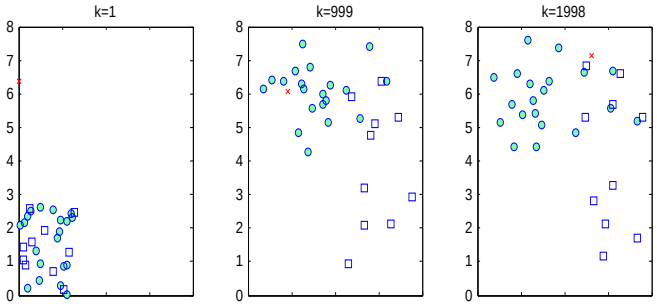

Fig. 2. Vehicle routing for a system with two types of vehicles, A (full circles) and B (empty squares). Only the reference points of the vehicles at the beginning of the periods are shown. Targets requiring service from type A appear with probability 40% and a distribution centered at $[2; 6]^T$. Targets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

obtain a first configuration using a deterministic deployment algorithm based on prior information about the environment, followed by the stochastic gradient scheme which accounts for the updated information collected by the robots. In general, the simplicity of the stochastic deployment algorithms would make 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 such a stochastic approach. For example, formation control using noisy observations and communication links can be studied from this point of view (the recent papers [30, 35, 20] study the related stochastic consensus problem using stochastic approximations).

REFERENCES

[1] W.F. Ames. *Numerical Methods for Partial Differential Equations*. Academic Press, 1977.

[2] 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.

[3] T. Balch and R.C. Arkin. Behavior-based formation control for multirobot teams. *IEEE Transactions on Robotics and Automation*, 14(6):926–939, 1998.

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

[5] D.J. Bertsimas and G.J. Van Ryzin. A stochastic and dynamic vehicle routing problem in the Euclidean plane. *Operations Research*, 39:601–615, 1991.

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

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

[8] 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.

[9] 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.

[10] J. Choi, S. Oh, and R. Horowitz. Distributed learning and cooperative control for multi-agent systems. *Automatica*, 45(12):2802–2814, December 2009.

[11] J. Cortés, S. Martinez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2):243–255, April 2004.

[12] S.P. Fekete, J.S.B. Mitchell, and K. Beurer. On the continuous Fermat-Weber problem. *Operations Research*, 53(1):61–76, January-February 2005.

[13] 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.

[14] V. Gazi and K.M. Passino. A class of attractions/repulsion functions for stable swarm aggregations. *International Journal of Control*, 77(18):1567–1579, 2004.

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

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

[17] R.M. Gray. Vector quantization. *IEEE ASSP Magazine*, 1(2):4–29, 1984.

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

[19] A. Howard, M.J. Mataric, and G.S. 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.

[20] M. Huang and J.H. Manton. Stochastic consensus seeking with noisy and directed inter-agent communication: Fixed and randomly varying topologies. *IEEE Transactions on Automatic Control*, 55(1):235–241, January 2010.

[21] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research*, 5(1):90–98, 1986.

[22] J. Kiefer and J. Wolfowitz. Stochastic estimation of the maximum of a regression function. *Annals of Mathematical Statistics*, 23(3):462–466, September 1952.

[23] T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59–69, 1982.

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

[25] L. Ljung. Analysis of recursive stochastic algorithms. *IEEE Transactions on Automatic Control*, 22(4):551–575, Augus 1977.

[26] J. Mac Queen. Some methods for the classification and analysis of multivariate observations. In L.N. Le Cam and J. Neyman, editors, *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297. University of California Press, 1967.

[27] P. Ogren, E. Fiorelli, and N.E. Leonard. Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment. *IEEE Transactions on Automatic Control*, 49(8):1292–1302, August 2004.

[28] R. Olfati-Saber. Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Transactions on Automatic Control*, 51(3):401–420, March 2006.

[29] K.M. Passino. Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems Magazine*, 22(3):52–67, 2002.

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

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

[32] W. Rudin. *Real and Complex Analysis*. McGraw-Hill Science, 3rd edition, 1986.

[33] F.E. 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*, pages 680–685, 2003.

[34] M. Schwager, D. Rus, and J.-J. Slotine. Decentralized, adaptive coverage control for networked robots. *The International Journal of Robotics Research*, 28(3):357–375, 2009.

[35] S.Kar and J.M.F. Moura. Distributed consensus algorithms in sensor networks with imperfect communication: Link failures and channel noise. *IEEE Transactions on Signal Processing*, 57(1):355–369, January 2009.

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