Multi-Robot Coordination for Estimation and Coverage of Unknown Spatial Fields

Alessia Benevento[1], María Santos[2], Giuseppe Notarstefano[3], Kamran Paynabar[4], Matthieu Bloch[2], and Magnus Egerstedt[2]

***Abstract*— We present an algorithm for multi-robot coverage of an initially unknown spatial scalar field characterized by a density function, whereby a team of robots simultaneously estimates and optimizes its coverage of the density function over the domain. The proposed algorithm borrows powerful concepts from Bayesian Optimization with Gaussian Processes that, when combined with control laws to achieve centroidal Voronoi tessellation, give rise to an adaptive sequential sampling method to explore and cover the domain. The crux of the approach is to apply a control law using a surrogate function of the true density function, which is then successively refined as robots gather more samples for estimation. The performance of the algorithm is justified theoretically under slightly idealized assumptions, by demonstrating asymptotic no-regret with respect to the coverage obtained with a known density function. The performance is also evaluated in simulation and on the Robotarium with small teams of robots, confirming the good performance suggested by the theoretical analysis.**

I. INTRODUCTION

Coverage of an environment constitutes a common task of multi-robot systems, whereby communicating robots coordinate their locations to achieve optimal coverage of a given area. Coverage is measured in terms of the robots' ability to collectively capture events whose importance is described by the density function of a scalar field [1]–[3]. Solutions to the coverage control problem find direct applications to autonomous environmental monitoring [4] or autonomous search-and rescue missions [5]. While most studies have concentrated on the coverage control problem with a known density function [6]–[8] or the estimation of an unknown density function [4], [9], follow-up works have only recently attempted to investigate the joint estimation and coverage of a domain in presence of an a priori unknown density function [5], [10]–[13].

Two previous approaches are particularly relevant to the present work. The first approach [10], [11] consists in assuming that the unknown density function can be parametrized as a linear combination of known basis functions. Consequently, the robots have to estimate the coefficients of the linear combination through sampling while attempting to achieve coverage. Although a complete protocol with provable guarantees has been proposed to achieve coverage, the protocol requires noiseless sensor observations and proceeds in a sequential manner, first estimating parameters and then running a distributed coverage control. The second approach [12], [13] consists in applying concepts from Bayesian Optimization and modeling the unknown density as the realization of a Gaussian Process [14]. This approach allows one to encourage simultaneous exploration and exploitation, and numerical results have demonstrated the good performance of a distributed adaptive coverage control strategy with online density function modeling through a mixture of Gaussian Processes.

In this paper, we further investigate the Bayesian approach to coverage control by leveraging recent advances in the regret analysis of Bayesian optimization problems. Similar to [12], [13], we propose an algorithm that combines the coverage control law of [1] with the powerful Gaussian Process Upper Confidence Bound (GP-UCB) algorithm [15] to create a surrogate density that is refined by acquiring more samples over time. However, a major difference between [12], [13] and the present work is that we take full advantage of some theoretical guarantees offered by Bayesian optimization developed over the last few years and we propose an algorithm that, under slightly idealized assumptions, offers asymptotically no-regret compared to the situation with a known density function. Despite high-level conceptual similarities with the algorithms proposed in [12], [13], our algorithm exhibits a few subtle differences that prove crucial to establish theoretical guarantees.

The remainder of the paper is organized as follows. In Section II, we formally introduce the optimal coverage problem. In Section III, we describe the proposed probabilistic model. In Section IV, we describe the proposed Prediction Correction Coverage (PCC) algorithm for the estimation and the coverage of spatial fields and demonstrate its asymptotic no-regret under slightly restrictive assumptions. In Section V, we corroborate the theoretical analysis in simulation and on the Robotarium.

This work was partially supported by the Army Research Lab through Grant No. DCIST CRA W911NF-17-2-0181 and by the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (grant agreement No 638992 - OPT4SMART).

[1]A. Benevento is with the Department of Engineering, University of Salento, via per Monteroni, 73100, Lecce, Italy, alessia.benevento@unisalento.it

[2]M. Santos, M. Bloch, and M. Egerstedt are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA {maria.santos,matthieu,magnus}@gatech.edu

[3]G. Notarstefano is with Department of Electrical, Electronic and Information Engineering, University of Bologna, Viale del Risorgimento, 2, 40136, Bologna, Italy, giuseppe.notarstefano@unibo.it

[4]K. Paynabar is with the School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA, kamran.paynabar@isye.gatech.edu

II. Preliminaries

We consider n mobile robots with sensing capability operating in an area described by a convex compact set $\mathcal{D} \subset \mathbb{R}^2$. The position of each robot $i \in [\![1, n]\!]$ is denoted by $x_i \in \mathbb{R}^2$, and positions are collectively grouped into a vector $\mathbf{x} = [x_1^\intercal, \cdots, x_n^\intercal]^\intercal \in \mathbb{R}^{2n}$. An event occurs in the area according to an a priori unknown event density function $\phi : \mathbb{R}^2 \to \mathbb{R}^+$. By communicating their positions $\mathbf{x}$ with their neighbors, the robots can compute a Voronoi partition $\{\mathcal{V}_i\}_{i=1}^n$ of the space $\mathcal{D}$, such that for all $i \in [\![1, n]\!]$

$$\mathcal{V}_i \triangleq \{q \in \mathbb{R}^2 : \forall j \neq i \quad \|q - x_i\|_2 \leqslant \|q - x_j\|_2\}. \tag{1}$$

Voronoi cell $\mathcal{V}_i$ can be viewed as the region of dominance of robot i, in which it is responsible for covering the domain. The centroid of each voronoi cell $\mathcal{V}_i$ is defined as

$$c_i(\phi) \triangleq \frac{1}{\int_{\mathcal{V}_i} \phi(q) dq} \int_{\mathcal{V}_i} q\phi(q) dq. \tag{2}$$

The set of locations $\{c_i(\phi)\}_{i=1}^n$ is known as a *centroidal Voronoi tessellation* (CVT) with respect to the density function ϕ. The objective is to design a distributed control law and a communication scheme such that the robots minimize the so-called *locational cost* [1]

$$\ell(\phi, \mathbf{x}) \triangleq \sum_{i=1}^n \int_{q \in \mathcal{V}_i} \|q - x_i\|_2^2 \phi(q) dq. \tag{3}$$

The locational cost captures the need for the robots to collectively move in positions that capture the relative areas of importance of the event, as defined by $\phi(\cdot)$. When the density function $\phi(\cdot)$ is known, a local minimizer (not necessarily unique) of the locational cost consists of the *centroids* of the Voronoi cells, which can be reached using a continuous-time version of Lloyd's algorithm [16].

We assume in this work that the density function is *unknown*; however, when in position x_i, robot i has the ability to sense the event and to obtain a noisy measurement of the density function according to the sensing model

$$y = \phi(x_i) + \varepsilon, \tag{4}$$

where $\varepsilon \sim \mathcal{N}(0, \sigma^2)$. We assume that the measurement noise ε is independent and identically distributed (i.i.d.) across time and space.

Since the proposed algorithm operates over T steps indexed by $t \in [\![1, T]\!]$, we now introduce some useful notation. All the variables related to step t are affixed with a superscript $(\cdot)^{(t)}$. In particular, each robot $i \in [\![1, n]\!]$ occupies location $x_i^{(t)} \in \mathbb{R}^2$ at step t and all locations are regrouped in a vector $\mathbf{x}^{(t)} \triangleq \left[x_1^{(t)}, \cdots, x_n^{(t)}\right]^\intercal \in \mathbb{R}^{2n}$. Each robot is then able to obtain a measurement $y_i^{(t)} = \phi(x_i^{(t)}) + \epsilon_i^{(t)}$, and all measurements are regrouped in a vector $\mathbf{y}^{(t)} \triangleq [y_1^{(t)}, \cdots, y_n^{(t)}]^\intercal \in \mathbb{R}^n$. It is also sometimes convenient to collect together the positions and measurements of the robots across consecutive time steps. We indicate this by replacing the superscript $(\cdot)^{(t)}$ by a superscript with the range of time steps, e.g., $(\cdot)^{(1:t)}$. In particular, all the positions of the robots during the algorithm are regrouped in a vector $\mathbf{x}^{(1:T)} \in \mathbb{R}^{2nT}$, and their corresponding measurements in a vector $\mathbf{y}^{(1:T)} \in \mathbb{R}^{nT}$.

III. Bayesian Model for Gaussian Processes

The key idea behind the proposed algorithm is to model the unknown density function $\phi(\cdot)$ as the realization of a Gaussian Process. Gaussian Process models are widely used for function estimation because of the flexibility they offer in modeling nonlinearities and thanks to the simple estimation of their associated parameters. A Gaussian Process is fully specified by a mean function $\mu : \mathcal{D} \to \mathbb{R}$ and a covariance function $k : \mathcal{D} \times \mathcal{D} \to \mathbb{R}$ [14]. There are many possible choices of mean and covariance functions [14]. In order to keep the model general, we only assume that the structures of μ and k are known up to certain hyper-parameters ρ and τ as

$$\mu = \mu(\cdot; \rho) \quad k = k(\cdot, \cdot; \tau),$$

and are continuous in those parameters. The proposed algorithm therefore only assumes that

$$\phi(\cdot) \sim \mathcal{GP}(\mu(\cdot; \rho), k(\cdot, \cdot; \tau)).$$

In particular, the experimental results in Section V are based on using a *squared-exponential kernel* for the covariance

$$k(x, x'; \tau) \triangleq \exp\left(-\frac{\|x - x'\|_2^2}{2\tau^2}\right), \tag{5}$$

where τ is a hyper-parameter known as the characteristic length scale, and a linear model for the mean function

$$\mu(x; \rho) \triangleq \rho^\intercal x. \tag{6}$$

Notice that (5) and (6) can model nonlinear functions. In particular, (6) should not be confused with (12) describing the mean of the posterior distribution of $\phi(\cdot)$. To deal with the presence of unknown hyper-parameters, we propose to follow an Empirical Bayes approach [17] by which we alternate between (i) the estimation of the hyper-parameters from collected observations; and (ii) Bayesian inference assuming that the estimated hyper-parameters are the true ones.

Specifically, at each step $t \in [\![1, T]\!]$ of the proposed algorithm, the observations $\mathbf{y}^{(1:t)}$ at locations $\mathbf{x}^{(1:t)}$ accumulated thus far by the n robots are used to perform Maximum Likelihood (ML) estimation of the hyper-parameters as

$$(\rho, \tau) = \operatorname*{argmax}_{\rho, \tau} p(\mathbf{y}^{(1:t)} | \mathbf{x}^{(1:t)}; \rho, \tau). \tag{7}$$

Conveniently, the density $p(\mathbf{y}^{(1:t)} | \mathbf{x}^{(1:t)}; \rho, \tau)$ is known in closed form with log-likelihood function

$$\begin{aligned} &\log(p(\mathbf{y}^{(1:t)} | \mathbf{x}^{(1:t)}; \rho, \tau)) = \\ &-\frac{1}{2}(\boldsymbol{\mu}^{(1:t)}(\rho) - \mathbf{y}^{(1:t)})^\intercal (\mathbf{K}_{\mathbf{x}^{(1:t)}}(\tau) + \sigma^2 \mathbf{I})^{-1} (\boldsymbol{\mu}^{(1:t)}(\rho) - \mathbf{y}^{(1:t)}) \\ &\qquad - \frac{1}{2} \log |\mathbf{K}_{\mathbf{x}^{(1:t)}}(\tau) + \sigma^2 \mathbf{I}| - \frac{t}{2} \log(2\pi), \end{aligned} \tag{8}$$

where $\boldsymbol{\mu}^{(1:t)}(\rho) \in \mathbb{R}^{nt}$ is the vector of mean values at all points sampled

$$\boldsymbol{\mu}^{(1:t)}(\rho) \triangleq [\mu(x_1^{(1)};\rho)\cdots\mu(x_n^{(t)};\rho)]^\intercal, \tag{9}$$

$\mathbf{I}$ is the identity matrix, and $\mathbf{K}_{\mathbf{x}^{(1:t)}}(\tau) \in \mathbb{R}^{nt\times nt}$ is the covariance matrix with entries

$$k(x_i^{(j)}, x_{i'}^{(j')};\tau) \text{ for } i,i' \in [\![1,n]\!] \text{ and } j,j' \in [\![1,t]\!]. \tag{10}$$

The first term of the log-likelihood function can be viewed as the estimation based on the data, while the second one can be viewed as a complexity penalty term. The ML estimation of ρ and τ at step t is therefore given by

$$(\rho^{(t)},\tau^{(t)}) = \underset{\rho,\tau}{\operatorname{argmin}}[(\boldsymbol{\mu}^{(1:t)}(\rho)-\mathbf{y}^{(1:t)})^\intercal(\mathbf{K}_{\mathbf{x}^{(1:t)}}(\tau)+\sigma^2\mathbf{I})^{-1} \\ (\boldsymbol{\mu}^{(1:t)}(\rho)-\mathbf{y}^{(1:t)})] + \log|\mathbf{K}_{\mathbf{x}^{(1:t)}}(\tau)+\sigma^2\mathbf{I}| \tag{11}$$

Note that the maximum likelihood solution of (11) improves over time as more and more locations are sampled. We call such a computation a *correction* step since it corrects the possibly imprecise values of the hyper-parameters. Once the parameters $\rho^{(t)}$ and $\tau^{(t)}$ are available, $\mu(\cdot;\rho^{(t)})$ and $k(\cdot,\cdot;\tau^{(t)})$ are treated as the true mean and covariance of the Gaussian Process.

Given mean and covariance functions, the posterior function of $\phi(\cdot)$, is given by a multivariate Gaussian distribution. Specifically, given the noisy observations $\mathbf{y}^{(1:t)}$ and the known sampling locations $\mathbf{x}^{(1:t)}$, the posterior distribution of $\phi(\cdot)$ is again a Gaussian Process with mean $\mu^{(t)}(x)$, covariance $k^{(t)}(x,x')$, and variance $\sigma^{(t)}(x)$ given by [15]

$$\mu^{(t)}(x) = \mu(x;\rho^{(t)}) + \mathbf{k}(x;\mathbf{x}^{(1:t)})^\intercal(\mathbf{K}_{\mathbf{x}^{(1:t)}}(\tau^{(t)})+\sigma^2\mathbf{I})^{-1} \\ (\mathbf{y}^{(1:t)} - \boldsymbol{\mu}^{(1:t)}(\rho^{(t)})) \tag{12}$$

$$k^{(t)}(x,x') = k(x,x';\tau^{(t)}) \\ - \mathbf{k}(x;\mathbf{x}^{(1:t)})^\intercal(\mathbf{K}_{\mathbf{x}^{(1:t)}}(\tau^{(t)})+\sigma^2\mathbf{I})^{-1}\mathbf{k}(x';\mathbf{x}^{(1:t)}) \tag{13}$$

and $\sigma^{(t)}(x) = k^{(t)}(x,x)$, where $\mathbf{K}_{\mathbf{x}^{(1:t)}}(\tau^{(t)})$ is defined as in (10) and $\mathbf{k}(x;\mathbf{x}^{(1:t)}) \in \mathbb{R}^{nT}$ is the vector with entries

$$k(x_i^{(j)}, x;\tau^{(t)}) \text{ for } i \in [\![1,n]\!] \text{ and } j \in [\![1,t]\!]. \tag{14}$$

We call such a computation a *prediction* step since it predicts the value of the density function $\phi(\cdot)$ in yet unexplored locations.

Note that the described procedure requires a centralized communication method to suitably fuse the collected data coming from different robots. How to make the procedure decentralized is left as a topic for future work [17].

IV. Prediction-Correction Algorithm

We now propose an algorithm, dubbed the *Prediction-Correction Coverage (PCC) algorithm*, which governs the trajectories and sensing of the robots, eventually leading them to a CVT for the true density function $\phi(\cdot)$. The algorithm allows a team of robot to simultaneously estimate and cover a spatial field by alternating between a *prediction* step, allowing the multi-robot team to select new points to be sampled, and a *correction* step, allowing the multi-robot team to update their estimation of the hyper-parameters for the underlying $\phi(\cdot)$, as explained in Section III.

Prediction-Correction Coverage Algorithm PCC

```
Initialization:
Define a non decreasing sequence $\{\beta^{(t)}\}_{t=1}^{T}$.
Randomly select locations $\mathbf{x}^{(0)} = [x_1^{(0)} \cdots x_n^{(0)}]^\intercal$.
Compute $\rho^{(0)}$ and $\tau^{(0)}$ as per (11).
Compute $\mu^{(0)}(\cdot)$ and $\sigma^{(0)}(\cdot)$ as per (12)-(13).
for $t = 1 \dots T$ do
    Prediction
    Compute surrogate $\phi^{(t)}$ for all $q \in \mathcal{D}$ as
        $\phi^{(t)}(q) \triangleq \mu^{(t-1)}(q) - \sqrt{\beta^{(t)}}\sigma^{(t-1)}(q)$.
    Run control of [1] with $\phi^{(t)}$ for robots to reach
        $\mathbf{x}^{(t)} \triangleq \text{CVT}(\phi^{(t)})$.
    Move drone to location
        $\tilde{x}^{(t)} = \operatorname{argmax}_{x\in\mathcal{D}} \sigma^{(t-1)}(x)$.
    Sample at locations of robots to obtain measurements
        $\forall i \in [\![1,n]\!] \quad y_i^{(t)} = \phi(x_i^{(t)}) + \varepsilon_i^{(t)}$.
    Sample at location of drone to obtain measurement
        $\tilde{y}^{(t)} = \phi(\tilde{x}^{(t)}) + \tilde{\varepsilon}^{(t)}$.
    Update $\mu^{(t)}(\cdot)$ and $\sigma^{(t)}(\cdot)$ as per (12)-(13).
    Correction
    Compute $\rho^{(t)}$ and $\tau^{(t)}$ as per (11).
```

A. Algorithm description

The proposed algorithm is explicitly given in the table above and we provide here supporting information and additional discussion. We assume the existence of an $n+1$th robot that does not engage in the coverage of the domain and merely helps the others acquire one extra sample of the density function per step. We refer to this robot as the *drone* while we simply refer to the others as the *robots*. As will be clear from the algorithm and its analysis, the drone need not really exist and its role could be played by another robot at the cost of further movement in the domain. As will be seen in the analysis of Section IV-B, the introduction of the drone is merely for analytical reasons and the algorithm runs equally well without this step.

The algorithm requires a non-decreasing sequence $\{\beta^{(t)}\}_{t=1}^{T}$ that will be precisely justified in Section IV-B. For now, assume that we set for some $\delta \in [0,1]$

$$\beta^{(t)} \triangleq 2\log\left(|\mathcal{D}|\, t^2 \frac{\pi^2}{6\delta}\right), \tag{15}$$

where the domain $\mathcal{D}$ has been discretized. The mathematical meaning of δ will be clarified in Lemma IV.5. At the beginning of each iteration $t \in [\![1, T]\!]$, the robots update a surrogate density function

$$\phi^{(t)}(q) \triangleq \mu^{(t-1)}(q) - \sqrt{\beta^{(t)}}\sigma^{(t-1)}(q), \tag{16}$$

which they use as their current approximation of the true density function ϕ. Note the presence of a minus sign in (16), which subtly differs from [12], [13] but plays a crucial role in Section IV-B. The intuition behind the choice of the surrogate density function is that we seek to underestimate the true density function to "push" robots towards areas of higher importance. The positivity of function (16) is not guaranteed. However, simulations suggest that the method converges even if we do not take into account this issue. We defer the precise study of this aspect for future work.

Once the surrogate density function is computed, the robots move to the centroidal Voronoi tessellation $\{c_i(\phi^{(t)})\}_{i=1}^{n}$ using the control law of [1]. This operation is concisely described as

$$\mathbf{x}^{(t)} \triangleq \mathrm{CVT}(\phi^{(t)}). \tag{17}$$

Concurrently, the drone moves to the location in the domain where $\sigma^{(t-1)}$ is maximum according to

$$\tilde{x}^{(t)} = \operatorname*{argmax}_{x \in \mathcal{D}} \sigma^{(t-1)}(x). \tag{18}$$

We do not specify the exact method to create this trajectory, which is irrelevant to the performance of the proposed algorithm. Note that the main cost in performing these movements lies with the CVT, and the calculations required by the drone are already part of those performed by the robots.

Once the robots and the drone reach their locations, they collect new measurements according to

$$\forall i \in [\![1, n]\!] \quad y_i^{(t)} = \phi(x_i^{(t)}) + \varepsilon_i^{(t)} \tag{19}$$

$$\tilde{y}^{(t)} = \phi(\tilde{x}^{(t)}) + \tilde{\varepsilon}^{(t)} \tag{20}$$

These new measurements are finally used to update $\mu^{(t)}$ and $\sigma^{(t)}$ and perform a correction step for the hyper-parameters $\rho^{(t)}$ and $\tau^{(t)}$.

B. Approximate No-regret Analysis of Coverage Control

We now show that, under some assumptions that we precisely state, the regret between the optimal cost and the cost effectively obtained through the proposed algorithm is sub-linear in T. Specifically, the regret $r^{(t)}$ at stage $t \in [\![1, T]\!]$ is defined as the difference between the locational cost incurred with the current locations $\mathbf{x}^{(t)}$ and the optimal locational cost, i.e.,

$$r^{(t)} \triangleq \ell(\phi, \mathbf{x}^{(t)}) - \min_{\mathbf{x}} \ell(\phi, \mathbf{x}) \geqslant 0. \tag{21}$$

The cumulative regret is then $R_T \triangleq \sum_{t=1}^{T} r^{(t)}$. If one can show that R_T grows sub-linearly with T, then $\lim_{T\to\infty} R_T/T = 0$ and one establishes a so-called *asymptotic no-regret*. An asymptotic no-regret implies that the algorithm is close to optimal on average in the long run.

We emphasize that our regret analysis does not strictly hold in the setup described earlier in the paper. Specifically, we make the following three restricting assumptions.

Assumption IV.1. *The hyper-parameters ρ and τ are known.*

Assumption IV.2. *The domain $\mathcal{D}$ is discrete.*

Assumption IV.3. *The coverage control law [1] for CVT solves the optimization problem: $\mathbf{x} = argmin_{\tilde{\mathbf{x}}}\, \ell(\phi, \tilde{\mathbf{x}})$*

Assumption IV.1 greatly simplifies our proof but is likely not so critical and could be removed in the future [18]. Assumption IV.2 is not fundamentally problematic either since the results can be generalized to a compact domain [15] and since our simulations already discretize $\mathcal{D}$. Assumption IV.3 is, however, central to our analysis. It is know that the coverage control law of [1] only achieves *local* optima of the locational cost with no guarantee of *global* optimality. Nevertheless, we heuristically justify our assumption by viewing the local optimum reached by the algorithm as an "attracting" optimum that effectively plays the role of a reachable global optimum. Furthermore, our analysis would hold if the coverage control law were shown to be optimal, and extensive simulations suggest that the control law provides a very good solution if not an optimal one. The proposed approach also offers additional benefits:

- Should another control law be constructed to achieve improved coverage based on a surrogate density function, this new control law could be plugged into our protocol without changing the sampling procedure.
- The proposed protocol performance can be precisely evaluated along several dimensions, including how much storage and computation is required at every node. This provides a clean framework to start developing fair comparisons between various protocols.

The main result of this section is then as follows.

Theorem IV.4. *Let $\delta \in [0; 1]$ and $\beta^{(t)} = 2\log\left(|\mathcal{D}|\, t^2\pi^2/6\delta\right)$. Under Assumptions IV.1-IV.3, if the event density function $\phi(\cdot)$ is the realization of a Gaussian Process with mean $\mu(\cdot)$ and covariance function $k(\cdot,\cdot) \leqslant 1$, then running the PCC algorithm ensures that*

$$\mathbb{P}\left(\forall T \geqslant 1 \quad R_T \leqslant \sqrt{c_1(n+1)T\beta^{(T)}\gamma_T}\right) \geqslant 1 - \delta$$

where $c_1 \triangleq 8\,|\mathcal{D}|^2 \max_{x,x'} \|x - x'\|_2^4 / \log(1 + \sigma^{-2})$ and $\gamma_T \triangleq \max_{\mathcal{A}\subset\mathcal{D}:|\mathcal{A}|=T} I(\mathbf{y}_{\mathcal{A}}; \boldsymbol{\phi}_{\mathcal{A}})$ with $I(\mathbf{y}_{\mathcal{A}}; \boldsymbol{\phi}_{\mathcal{A}})$ the mutual information between the density and observations in locations in $\mathcal{A}$. For many kernels $\gamma_T = o(T)$, so that the PCC algorithm asymptotically has no-regret.

Proof: We first recall a result from [15, Lemma 5.6].

Lemma IV.5. *Pick $\delta \in [0; 1]$ and set $\beta^{(t)} = 2\log(|\mathcal{D}|\,\pi_t/\delta)$, where $\sum_{t\geqslant 1} \pi_t^{-1} = 1$, $\pi_t > 0$ (say $\pi_t = \frac{\pi^2 t^2}{6}$). Then,*

$$\left|\phi(x) - \mu^{(t-1)}(x)\right| \leqslant \sqrt{\beta^{(t)}}\sigma^{(t-1)}(x)$$

holds for any $x \in \mathcal{D}$ and any $t \geqslant 1$ with probability at least $1 - \delta$.

In the remaining of the proof, we assume that the high-probability event of Lemma IV.5 holds. We let $\ell^* \triangleq \min_{\mathbf{x}} \ell(\phi, \mathbf{x})$ and $\mathbf{x}^* \triangleq \operatorname{argmin}_{\mathbf{x}} \ell(\phi, \mathbf{x})$. By Assumption IV.3, we have

$$\ell(\phi^{(t)}, \mathbf{x}^{(t)}) \leqslant \ell(\phi^{(t)}, \mathbf{x}^*) \leqslant \ell^*, \tag{22}$$

where the last inequality follows because we are assuming that the event in Lemma IV.5 holds. The regret at step t is then

$$\begin{aligned} r^{(t)} &\triangleq \ell(\phi, \mathbf{x}^{(t)}) - \ell^* \overset{(a)}{\leqslant} \ell(\phi, \mathbf{x}^{(t)}) - \ell(\phi^{(t)}, \mathbf{x}^{(t)}) \\ &\overset{(b)}{=} \sum_i \sum_{q \in \mathcal{V}_i^{(t)}} \|q - x_i^{(t)}\|_2^2 \left(\phi(q) - \phi^{(t)}(q)\right) \\ &\overset{(c)}{\leqslant} 2\sqrt{\beta^{(t)}} \left(\sum_i \sum_{q \in \mathcal{V}_i^{(t)}} \left\| q - x_i^{(t)} \right\|_2^2 \right) \max_{q \in \mathcal{D}} \sigma^{(t-1)}(q) \\ &\overset{(d)}{\leqslant} 2c_0 \sqrt{\beta^{(t)}} \max_{q \in \mathcal{D}} \sigma^{(t-1)}(q) \triangleq 2c_0 \sqrt{\beta^{(t)}} \sigma^{(t-1)}(\tilde{x}^{(t)}), \end{aligned} \tag{23}$$

where (a) follows from (22), (b) follows by the definition of the locational cost, (c) follows because $\phi(q) \leqslant \mu^{(t-1)}(q) + \sqrt{\beta^{(t)}}\sigma^{(t-1)}(q)$ and $\phi^{(t)}(q) = \mu^{(t-1)}(q) - \sqrt{\beta^{(t)}}\sigma^{(t-1)}(q)$, and (d) follows by setting $c_0 = |\mathcal{D}| \max_{x, x' \in \mathcal{D}} \|x - x'\|_2^2$. Note that c_0 only depends on the geometry of the domain $\mathcal{D}$ and not on the number of robots n. One might hope to develop a bound that *decreases* with the number of robots, but this is left for future work.

Following standard steps [15, Lemma 5.4] and using the fact that $\beta^{(t)}$ is non-decreasing with t, we obtain

$$\left(r^{(t)}\right)^2 \leqslant 4c_0^2 \beta^{(t)} \left(\sigma^{(t-1)}(\tilde{x}^{(t)})\right)^2 \tag{24}$$

$$\leqslant c_1 \beta^{(T)} \frac{1}{2} \log \left(1 + \left(\frac{\sigma^{(t-1)}(\tilde{x}^{(t)})}{\sigma} \right)^2 \right) \tag{25}$$

for $c_1 = 8c_0^2 / \log(1 + \sigma^{-2})$. Therefore,

$$\sum_{t=1}^{T} \left(r^{(t)}\right)^2 \leqslant c_1 \beta^{(T)} \sum_{t=1}^{T} \log \left(1 + \frac{\sigma^{(t-1)}(\tilde{x}^{(t)})^2}{\sigma^2} \right).$$

We now develop an information gain bound. For $t \in \llbracket 1, T \rrbracket$, we slightly abuse notation and define $\phi(\mathbf{x}^{(1:t)}) \in \mathbb{R}^{nt}$ as

$$\phi(\mathbf{x}^{(1:t)}) \triangleq \left[\; \phi(x_1^{(1)}) \cdots \phi(x_n^{(1)}) \cdots \phi(x_1^{(t)}) \cdots \phi(x_n^{(t)}) \; \right]^{\mathsf{T}}.$$

Similarly we define $\phi(\tilde{x}^{(1:t)}) \in \mathbb{R}^t$ as

$$\phi(\tilde{x}^{(1:t)}) \triangleq \left[\; \phi(\tilde{x}^{(1)}) \cdots \phi(\tilde{x}^{(t)}) \; \right]^{\mathsf{T}}.$$

Recall that for every $i \in \llbracket 1, n \rrbracket$ and every $t \in \llbracket 1, T \rrbracket$, we have

$$y_i^{(t)} \triangleq \phi(x_i^{(t)}) + \varepsilon_{i,t} \qquad \tilde{y}^{(t)} = \phi(\tilde{x}^{(t)}) + \tilde{\varepsilon}_t \tag{26}$$

where all the additive noises are $\mathcal{N}(0, \sigma^2)$ and i.i.d. Then,

$$\begin{aligned} &I(\phi(\mathbf{x}^{(1:T)}), \phi(\tilde{x}^{(1:T)}); \mathbf{y}^{(1:T)}, \tilde{y}^{(1:T)}) \\ &= h(\mathbf{y}^{(1:T)}, \tilde{y}^{(1:T)}) - h(\mathbf{y}^{(1:T)}, \tilde{y}^{(1:T)} | \phi(\mathbf{x}^{(1:T)}), \phi(\tilde{x}^{(1:T)})) \\ &= h(\mathbf{y}^{(1:T)}, \tilde{y}^{(1:T)}) - \frac{(n+1)T}{2} \log\left(2\pi e \sigma^2\right), \end{aligned} \tag{27}$$

where the last equality follows by definition of the sensing model (26). By the chain rule of differential entropy, we have

$$h(\mathbf{y}^{(1:T)}, \tilde{y}^{(1:T)}) = \sum_{t=1}^{T} h(\mathbf{y}^{(t)}, \tilde{y}^{(t)} | \mathbf{y}^{(1:t-1)}, \tilde{y}^{(1:t-1)}) \tag{28}$$

with $h(\mathbf{y}^{(t)}, \tilde{y}^{(t)} | \mathbf{y}^{(1:t-1)}, \tilde{y}^{(1:t-1)})$

$$= h(\tilde{y}^{(t)} | \mathbf{y}^{(1:t-1)}, \tilde{y}^{(1:t-1)}) + h(\mathbf{y}^{(t)} | \mathbf{y}^{(1:t-1)}, \tilde{y}^{(1:t)}). \tag{29}$$

We analyze each term in (29) independently. With the proposed algorithm, the sequence of sensing points $\mathbf{x}^{(1:t)}$ and $\tilde{x}^{(1:t)}$ is a *deterministic* function of $\mathbf{y}^{(1:t-1)}$ and $\tilde{y}^{(1:t-1)}$. Therefore,

$$\begin{aligned} &h(\tilde{y}^{(t)} | \mathbf{y}^{(1:t-1)}, \tilde{y}^{(1:t-1)}) \\ &\qquad = h(\tilde{y}^{(t)} | \mathbf{x}^{(1:t)}, \tilde{x}^{(1:t)}, \mathbf{y}^{(1:t-1)}, \tilde{y}^{(1:t-1)}) \\ &\qquad = \frac{1}{2} \log \left(2\pi e (\sigma^2 + \sigma^{(t-1)}(\tilde{x}^{(t)})^2) \right), \end{aligned} \tag{30}$$

where the last equality follows since $\tilde{y}^{(t)}$ conditioned on $\mathbf{x}^{(1:t)}, \tilde{x}^{(1:t)}, \mathbf{y}^{(1:t-1)}, \tilde{y}^{(1:t-1)}$ is distributed according to $\mathcal{N}(\mu^{(t-1)}(\tilde{x}^{(t)}), \sigma^2 + \sigma^{(t-1)}(\tilde{x}^{(t)})^2)$ and $\sigma^{(t-1)}(\tilde{x}^{(t)})$ does not depend on $\mathbf{y}^{(1:t-1)}, \tilde{y}^{(1:t-1)}$. Similarly,

$$\begin{aligned} &h(\mathbf{y}^{(t)} | \mathbf{y}^{(1:t-1)}, \tilde{y}^{(1:t)}) \\ &\qquad = h(\mathbf{y}^{(t)} | \mathbf{x}^{(1:t)}, \tilde{x}^{(1:t)}, \mathbf{y}^{(1:t-1)}, \tilde{y}^{(1:t)}) \\ &\qquad = \frac{1}{2} \log \left| 2\pi e \left(\sigma^2 \mathbf{I} + \mathbf{\Sigma}_t\right) \right| \end{aligned} \tag{31}$$

where the last equality follows since $\mathbf{y}^{(t)}$ conditioned on $\mathbf{x}^{(1:t)}, \tilde{x}^{(1:t)}, \mathbf{y}^{(1:t-1)}, \tilde{y}^{(1:t)}$ is distributed according to $\mathcal{N}(\tilde{\boldsymbol{\mu}}_t, \mathbf{\Sigma}_t(\mathbf{x}^{(1:t)}))$ with some $\tilde{\mu} \in \mathbb{R}^n$ and $\mathbf{\Sigma}^{(t)}(\mathbf{x}^{(1:t)}) \in \mathbb{R}^{n \times n}$ computed appropriately. Combining (27)-(31) and using the non-negativity of mutual information, we obtain

$$\begin{aligned} &I(\phi(\mathbf{x}^{(1:T)}), \phi(\tilde{x}^{(1:T)}); \mathbf{y}^{(1:T)}, \tilde{y}^{(1:T)}) \\ &= \sum_{t=1}^{T} \left[\frac{1}{2} \log \left(1 + \frac{\sigma^{(t-1)}(\tilde{x}^{(t)})^2}{\sigma^2} \right) + \frac{1}{2} \log \left| I + \frac{1}{\sigma^2} \mathbf{\Sigma}_t \right| \right] \\ &\geqslant \sum_{t=1}^{T} \frac{1}{2} \log \left(1 + \frac{\sigma^{(t-1)}(\tilde{x}^{(t)})^2}{\sigma^2} \right). \end{aligned} \tag{32}$$

The exact behavior of $I(\phi(\mathbf{x}^{(1:T)}), \phi(\tilde{x}^{(1:T)}); \mathbf{y}^{(1:T)}, \tilde{y}^{(1:T)})$ is difficult to analyze because of how the sequence of robot positions evolve at each step of the algorithm. Fortunately, we can use the universal upper bound

$$\begin{aligned} &I(\phi(\mathbf{x}^{(1:T)}), \phi(\tilde{x}^{(1:T)}); \mathbf{y}^{(1:T)}, \tilde{y}^{(1:T)}) \\ &\overset{(a)}{=} I(\tilde{y}^{(1:T)}; \phi(\mathbf{x}^{(1:T)}), \phi(\tilde{x}^{(1:T)})) + \\ &\sum_{i=1}^{n} I(y_i^{(1:T)}; \phi(\mathbf{x}^{(1:T)}), \phi(\tilde{x}^{(1:T)}) | y_1^{(1:T)}, \ldots, y_{i-1}^{(1:T)}, \tilde{y}^{(1:T)}) \\ &\overset{(b)}{\leqslant} I(\tilde{y}^{(1:T)}; \phi(\tilde{x}^{(1:T)})) + \sum_{i=1}^{n} I(y_i^{(1:T)}; \phi(x_i^{(1:T)})) \\ &\overset{(c)}{\leqslant} (n+1) \max_{\mathcal{A} \subset \mathcal{D} : |\mathcal{A}| = T} I(\mathbf{y}_{\mathcal{A}}; \boldsymbol{\phi}_{\mathcal{A}}), \end{aligned} \tag{33}$$

where (a) follows by the chain rule of mutual information; (b) follows because conditioning does not increase entropy; and (c) follows trivially upon defining $\mathbf{y}_{\mathcal{A}}$ and $\boldsymbol{\phi}_{\mathcal{A}}$ appropriately [15]. Upon setting $\gamma_T \triangleq \max_{\mathcal{A}\subset\mathcal{D}:|\mathcal{A}|=T} I(\mathbf{y}_{\mathcal{A}};\boldsymbol{\phi}_{\mathcal{A}})$ and combining (25), (32), and (33), we obtain

$$\sum_{t=1}^{T}\left(r^{(t)}\right)^2 \leqslant c_1\beta^{(T)}(n+1)\gamma_T,$$

so that with a direct application of Cauchy-Schwarz inequality

$$\sum_{t=1}^{T} r^{(t)} \leqslant \sqrt{c_1(n+1)T\beta^{(T)}\gamma_T}.$$

From [19, Theorem 8], we know that γ_T scales sub-linearly with T for many kernels. In particular, for the squared exponential kernel, we have $\gamma_T = \mathcal{O}([\log(T^3\log T)]^3)$.

V. Experimental results

The performance of the proposed algorithm was evaluated in simulation and on the Robotarium [20], a remotely accessible multi-robot testbed at the Georgia Institute of Technology. The Robotarium is populated by small-scale differential-drive robots that move according to a script uploaded by the web.

In all experiments, the robots start from a randomly chosen set of positions $\mathbf{x}^{(0)}$ and the standard deviation of the observation noise is $\sigma = 0.01$. The value of δ in (15) is set to 0.01. The algorithm was tested with teams of four and six robots and with the following density function:

$$\phi(x) = 3(1-x_1)^2 e^{-x_1^2-(x_2+1)^2} - \frac{1}{3}e^{-(x_1+1)^2-x_2^2} + \frac{7}{2} - 10\left(\frac{x_1}{5} - x_1^3 - x_2^5\right)e^{-(x_1^2-x_2^2)}. \quad (34)$$

The mean and covariance of the Gaussian Process are chosen to be a squared exponential kernel and a linear model as per (5) and (6), respectively.

The trajectories followed by a team of four robots in the Robotarium and in simulation are shown in Fig. 1 and Fig. 2, respectively. In each figure, the right panel displays the trajectories that robots would have followed had they known the true density function, while the left panel shows the actual trajectories followed with the proposed algorithm. As expected, the trajectories do not sample the entire domain and take advantage of the *prediction* step in the algorithm to only select significant new sampling points.

To develop additional insight into the benefits of the proposed algorithm, the evolution of the locational cost $\ell(\phi, \mathbf{x}^{(t)})$ as the function of time with teams of four and six robots is shown in Fig. 3.The time instants when observations are obtained (corresponding to steps t in the algorithm) are indicated with markers ($\times$) but the locational cost is tracked at all times when robots are moving. While the cost is always slightly larger than the one incurred with known ϕ, the loss of performance vanishes with the number of iterations and confirms the ability of the proposed algorithm to efficiently explore while achieving good coverage. Figure 3 also shows the evolution of the locational cost when the robots move according to the GP-UCB algorithm [12]. While the GP-UCB algorithm approaches the cost of the algorithm under a known ϕ faster than the approach proposed in this paper, the GP-UCB algorithm does not offer theoretical guarantees.

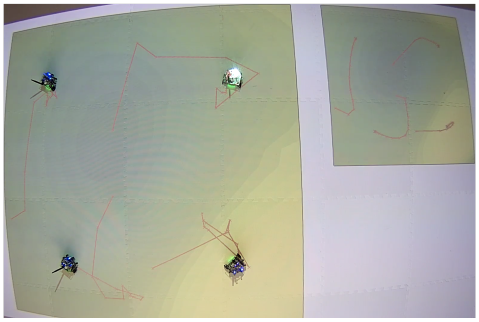

Fig. 1. Snapshot of the experiment performed with four robots when the true density function is ϕ.The right panel shows the projection of ϕ on the domain. The red lines represent the trajectories that the robots would have if they knew ϕ. The left panel shows the projection of the estimation of the function at the end of the experiment. The red lines represent the trajectories of the robots.

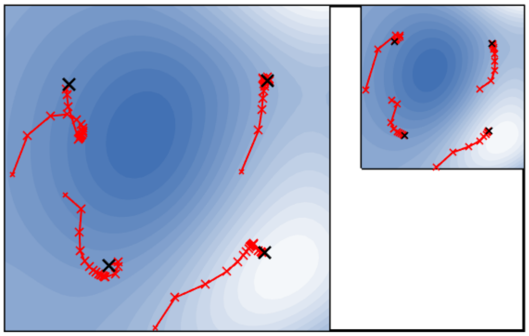

Fig. 2. Trajectories of four robots when the true density function is ϕ. The right panel shows the projection of ϕ on the domain. The red lines represent the trajectories that the robots would have if they knew ϕ. The left panel shows the projection of the estimation of the function at the end of the simulation. The red lines represent the trajectories assigned by the proposed algorithm.

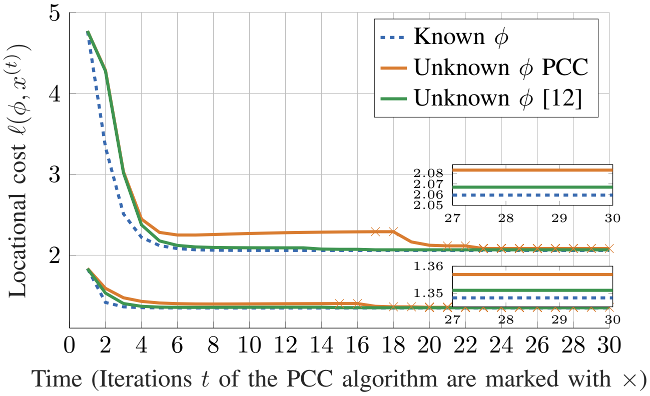

Fig. 3. Evolution of the locational cost for the density function ϕ. The group of lines above is the evolution of the cost in presence of four robots. The triplet of lines below is the evolution of it in presence of six robots.

References

[1] J. Cortes, 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, 2004.

[2] F. Bullo, J. Cortes, and S. Martinez, *Distributed control of robotic networks: a mathematical approach to motion coordination algorithms*. Princeton University Press, 2009, vol. 27.

[3] J. Cortés, S. Martinez, T. Karatas, and F. Bullo, “Coverage control for mobile sensing networks: Variations on a theme,” in *Mediterranean Conference on Control and Automation*. Lisbon, Portugal Lisbon, Portugal, 2002, pp. 9–13.

[4] A. Singh, A. Krause, and W. J. Kaiser, “Nonmyopic adaptive informative path planning for multiple robots,” in *Twenty-First International Joint Conference on Artificial Intelligence*, 2009.

[5] A. Carron, M. Todescato, R. Carli, L. Schenato, and G. Pillonetto, “Multi-agents adaptive estimation and coverage control using gaussian regression,” in *2015 European Control Conference (ECC)*. IEEE, 2015, pp. 2490–2495.

[6] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava, “Coverage problems in wireless ad-hoc sensor networks,” in *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No. 01CH37213)*, vol. 3. IEEE, 2001, pp. 1380–1387.

[7] A. Breitenmoser, M. Schwager, J.-C. Metzger, R. Siegwart, and D. Rus, “Voronoi coverage of non-convex environments with a group of networked robots,” in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 4982–4989.

[8] M. Santos, Y. Diaz-Mercado, and M. Egerstedt, “Coverage control for multirobot teams with heterogeneous sensing capabilities,” *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 919–925, 2018.

[9] G. A. Hollinger and G. S. Sukhatme, “Sampling-based motion planning for robotic information gathering.” in *Robotics: Science and Systems*, vol. 3, no. 5. Citeseer, 2013.

[10] 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.

[11] M. Schwager, M. P. Vitus, S. Powers, D. Rus, and C. J. Tomlin, “Robust adaptive coverage control for robotic sensor networks,” *IEEE Transactions on Control of Network Systems*, vol. 4, no. 3, pp. 462–476, Sep. 2017.

[12] W. Luo and K. Sycara, “Adaptive sampling and online learning in multi-robot sensor coverage with mixture of gaussian processes,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 6359–6364.

[13] W. Luo, C. Nam, G. Kantor, and K. Sycara, “Distributed environmental modeling and adaptive sampling for multi-robot sensor coverage,” in *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2019, pp. 1488–1496.

[14] C. E. Rasmussen and C. K. Williams, “Gaussian processes for machine learning,” 2006.

[15] N. Srinivas, A. Krause, S. Kakade, and M. Seeger, “Gaussian process optimization in the bandit setting: No regret and experimental design,” in *Proceedings of the 27th International Conference on Machine Learning*, ser. ICML’10. USA: Omnipress, 2010, pp. 1015–1022. [Online]. Available: http://dl.acm.org/citation.cfm?id=3104322.3104451

[16] S. Lloyd, “Least squares quantization in pcm,” *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.

[17] F. Sasso, A. Coluccia, and G. Notarstefano, “An empirical bayes approach for distributed estimation of spatial fields,” in *2018 European Control Conference (ECC)*. IEEE, 2018, pp. 2206–2211.

[18] F. Berkenkamp, A. P. Schoellig, and A. Krause, “No-regret bayesian optimization with unknown hyperparameters,” *Journal of Machine Learning Research*, vol. 20, no. 50, pp. 1–24, Feb. 2019. [Online]. Available: http://jmlr.org/papers/v20/18-213.html

[19] N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger, “Information-theoretic regret bounds for gaussian process optimization in the bandit setting,” *IEEE Transactions on Information Theory*, vol. 58, no. 5, pp. 3250–3265, 2012.

[20] S. Wilson, P. Glotfelter, L. Wang, S. Mayya, G. Notomista, M. Mote, and M. Egerstedt, “The robotarium: Globally impactful opportunities, challenges, and lessons learned in remote-access, distributed control of multirobot systems,” *IEEE Control Systems Magazine*, vol. 40, no. 1, pp. 26–44, Feb 2020.