

Learning a Spatial Field in Minimum Time With a Team of Robots

Varun Suryan¹, *Student Member, IEEE*, and Pratap Tokekar², *Member, IEEE*

Abstract—In this article, we study an informative path-planning problem where the goal is to minimize the time required to learn a spatially varying entity. We use Gaussian process (GP) regression for learning the underlying field. Our goal is to ensure that the GP posterior variance, which is also the mean square error between the learned and actual fields, is below a predefined value. We study three versions of the problem. In the placement version, the objective is to minimize the number of measurement locations while ensuring that the posterior variance is below a predefined threshold. In the mobile robot version, we seek to minimize the total time required to visit and collect measurements from the measurement locations using a single robot. We also study a multirobot version where the objective is to minimize the time required by the last robot to return to a common starting location called depot. By exploiting the properties of GP regression, we present constant-factor approximation algorithms. In addition to the theoretical results, we also compare the empirical performance using a real-world dataset, with other baseline strategies.

Index Terms—Gaussian Process (GP) regression, informative path planning (IPP).

I. INTRODUCTION

SENSING, modeling, and tracking various spatially varying entities can improve our knowledge and understanding of them. This can have significant economic, environmental, and health implications. For example, knowing the content of various nutrients in the soil of a farm can help the farmers better understand soil chemistry. Understanding soil chemistry is helpful for the farmers to improve the yield and reduce the application of fertilizers [1]. An overload of certain chemicals inside a water body may have a significant impact on marine life. Knowing the spatial extent of the spill is necessary for effective control and mitigation strategies [2]. Understanding the spatial variation of rock minerals can help in efficient mining strategies [3]. In all such applications, a key first step is the collection of data using appropriate sensors which can then be used to build models of the underlying phenomenon. However, collecting data can

Manuscript received March 6, 2020; accepted May 7, 2020. Date of publication June 17, 2020; date of current version October 1, 2020. This work was supported in part by the National Science Foundation under Grant 1637915 and in part by NIFA under Grant 2018-67007-28380. This article was recommended for publication by Associate Editor M. Schwager and Editor P. Robuffo Giordano upon evaluation of the reviewers' comments. (*Corresponding author: Varun Suryan.*)

The authors were with the Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA 24060, USA. They are now with the Department of Computer Science, University of Maryland, College Park, MD 20740, USA (e-mail: suryan@umd.edu; tokekar@umd.edu).

Color versions of one or more of the figures in this article are available online at <https://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TRO.2020.2994003



Fig. 1. Single quadcopter can fly over a farm and measure the height of the crop using a LIDAR sensor.

be tedious and often requires careful human planning. Manual data collection can also be dangerous. For example, volcano monitoring data help to see where previous lava flows have gone and previous ash fall has occurred. However, volcanic ash is usually pulverized rocks and glass particles and potentially catastrophic for the people engaged in monitoring [4]. One alternative which would alleviate the human risks of manual data collection is the use of robots equipped with appropriate sensors to collect data.

There are many factors to consider when deploying robots for data collection. Usually, a tradeoff must be made between the quantity of sensing resources (e.g., number of deployed robots, energy consumption, mission time) and the quality of data collected. The robots can be deployed to act as stationary or mobile sensors depending on the application (see Fig. 1). Deploying robots to function as mobile sensors is especially challenging because of the need for path planning. While deploying mobile robotic sensors, one needs to plan the most informative resource-constrained observation paths to minimize the uncertainty in modeling and tracking the spatial phenomena.

Planning informative resource-constrained observation paths for robot sensors to estimate a spatially varying entity, often known as informative path planning (IPP), has received recent attention in the robotics community [5]–[10]. IPP deals with the problem of deciding an autonomous robot path along which maximum possible information about a quantity of interest can be extracted while operating under a set of resource constraints. In this article, our quantity of interest is a spatially varying phenomenon, often represented by a spatial field.¹ Generally, the underlying spatial field is specified by a probabilistic model. One of the commonly used probabilistic models is Gaussian process

¹In this article, a spatial field is a function $f(x)$, $x \in U$, that is defined over a spatial domain, $U \subset \mathbb{R}^2$.

(GP) [11]. GPs provide an mathematically convenient way of performing nonparametric regression while making fewer assumptions on the underlying field. They allow for expressing domain knowledge through the choice of kernel functions. In particular, for spatially varying fields, numerous studies have shown the efficacy of modeling with GPs [12]. An alternative would be geometric models which make strong assumptions and cannot represent the stochastic noise in the measurements directly [5]. Thus, probabilistic models make a suitable candidate for such scenarios.

Once the underlying spatial field is modeled, the next task is to plan a robot path based on that model. The robot travels along a path planned in this step. Several metrics can be used to perform the planning step. An information-theoretic metric, such as mutual information (MI), entropy, or variance, is typically used as a criterion to drive the robot to sampling locations [13]. Generally, the information-theoretic metrics are submodular and hence, an approximation guarantee can be given on the performance of the resulting algorithms [5]. Unfortunately, the information-theoretic metrics, such as entropy, MI, etc., are indirect and do not consider the accuracy of the predictions. Unlike these works, we study how to ensure that the GP predicted mean² is accurate and present a constant-factor approximation algorithm if the hyperparameters of the GP kernel do not change.

We use variance of GP prediction as the metric to perform the planning step. Predictive variance also turns out to be the mean square error (MSE) in GP prediction if the hyperparameters are known [14]. Our goal in this work is to plan the informative paths such that the predictive variance at all locations is below a predefined threshold Δ , after collecting measurements using mobile sensors. This leads to same guarantees on MSE as well. We study the following three related problems, that of:

- 1) finding measurement locations to make measurements;
- 2) planning a tour for a single robot to visit those measurement locations;
- 3) planning tours for multiple mobile robots.

This to ensure that the predictive variance is below Δ . The objective is to minimize the number of measurement locations in the first problem and the total tour time in the second problem. With multiple robots, the objective is to minimize the maximum time taken among all the robots. The total tour time is given by the measurement time and the travel time between measurement locations. The measurement time depends on the number of measurements taken at each location as well as the time to take a single measurement. Depending on the sensor, the measurement time can be zero (e.g., cameras) or nonzero (e.g., soil probes measuring organic content). We show that a nonadaptive algorithm suffices to solve the problem and yields a polynomial-time constant-factor approximation to the optimal algorithm. While other algorithms have been proposed before for estimating spatial fields, this is the first result that provides the theoretical guarantees on the total time for ensuring predictive accuracy at all points. Our main contributions include the following:

- 1) introducing stationary sensor placement and mobile sensor algorithms for ensuring that the predictive variance, and hence MSE, at each location in the environment is below a predefined threshold;
- 2) providing polynomial-time constant-factor approximation guarantees on their performance;
- 3) showing their performance on a real-world dataset comprising of OM concentrations at various locations within a farm.

Similar problems have been studied in the literature. For example, Yfantis *et al.* [15] studied a stationary sensor problem. Their approach considers and investigates only three types of predefined placement designs while for a general case none of them may be a good design. The algorithms presented in this work are not restricted to any predefined placement strategy. Further, we are not aware of any existing theoretical guarantees on the mobile sensor problems studied in this article.

The rest of the article is organized as follows: In Section II, we present a discussion on related works and background on the problems studied. In Section III, we formally present the problems and their solutions. Simulation results are presented in the Section IV. Section V concludes this article.

A preliminary version of this work was presented at the 13th International Workshop on the Algorithmic Foundations of Robotics (WAFR'18) [16]. In the preliminary version, we provided guarantees for the chance constraints of incorrect predictions using an aggregate measure of prediction error. In current work, a more direct performance criterion, MSE at each location of the environment, is considered which leads to stronger theoretical guarantees. Also, an extension of the algorithms for the multirobot case is presented.

II. RELATED WORK AND BACKGROUND

We begin by reviewing the related work in sensor placement where the goal is to cover a given environment using sensors placed at fixed locations and mobile sensing where sensors can move and collect measurements from different locations.

A. Stationary Sensor Placement

When monitoring a spatial phenomenon, such as temperature or humidity in an environment, selection of a limited number of sensors and their locations is an important problem. The goal in this problem is to select the best k out of n possible sensor locations and use the measurements from these to make predictions about the spatial phenomenon. The typical formulation of a sensor selection problem makes it NP-hard [17]. Previous work used global optimization techniques such as branch and bound to exactly solve this problem [18], [19]. However, these exact approaches are often computationally intensive.

One can solve the task as an instance of the art-gallery problem [20], [21]—find the minimum set of guards inside a polygonal workspace from which the entire workspace is visible. However, this version of the problem only covers vision-based sensors and does not consider noisy measurements [5].

An alternative approach from spatial statistics is to learn a model of the phenomenon, typically as a GP [22], [23]. The

²We use *predicted* mean and *estimated* mean interchangeably since the function is independent of time.

learned GP model can then be used to predict the effect of placing sensors at locations, and thus, optimize their placement. For a given GP model, many criteria including information-theoretic ones have been proposed to evaluate the quality of placement. Shewry and Wynn [24] introduced the maximum entropy criterion where the sensors are placed sequentially at the locations of highest entropy. Ko *et al.* [25] proposed a greedy algorithm by formulating the entropy maximization as maximizing the determinant of the covariance matrix. However, the entropy criterion tends to place the sensors at the boundary of the environment thus wasting sensed information [26]. MI can be used as well [22], [27], [28]. Krause *et al.* [5] studied the problem of maximizing MI for optimizing sensor placement problem. They presented a polynomial-time approximation algorithm with constant factor guarantee by exploiting submodularity [29]. Eventually, they showed that MI criterion leads to improved accuracy with a fewer number of sensors compared to other common design criteria such as entropy [24], A-optimal, D-optimal, and E-optimal design [30].

The abovementioned methods estimate the prediction error indirectly. Nguyen *et al.* [31] considered choosing a set of n potential sensor measurements such that the root mean square prediction error is minimized. They presented an annealing-based algorithm for the sensor selection problem. Their algorithm started by selecting a potential subset of cardinality k from the entire population of sensor locations. After that, it iteratively attempted to substitute the members of the selected subset by its neighbors according to an optimization criterion.

None of the criteria discussed above cannot directly make any guarantees on the MSE in predictions at each point in the environment. Instead, we design a sensor placement algorithm which results in an accurate reconstruction of the spatial field using the collected sensor measurements. Most works in the past have focused on optimizing an objective function (entropy, MI, etc.) given the resource constraints (limited energy, number of sensors, and time, etc.). We optimize the resource requirement given the objective constraint (MSE below a predefined threshold Δ), predictive accuracy more than a predefined threshold in our case.

B. Mobile Sensing

The goal in the mobile sensing problem, also known as IPP, is to compute paths for robots acting as mobile sensors. Paths are being computed in order to accurately estimate some underlying phenomenon, typically a spatial field [32], [33]. A central problem in IPP is to identify the hotspots in a large-scale spatial field. Hotspots are the regions in which the spatial field measurements exceed a predefined threshold. In many applications, it is necessary to assess the spatial extent and shape of the hotspot regions accurately. Low *et al.* [34] presented a decentralized active robotic exploration strategy for probabilistic classification/labeling of hotspots in a GP-based spatial field. The time needed by their strategy is independent of the map resolution and the number of robots, thus making it practical for *in situ*, real-time active sampling. Another formulation in hotspot identification is that of level set identification [35].

Previous works on level set boundary estimation and tracking [36]–[38] have primarily focused on communication of the sensor nodes, without giving much attention to individual sampling locations. Bryan *et al.* [39] proposed the straddle heuristic, which selects sampling locations by trading off uncertainty and proximity to the desired threshold level, both estimated using GPs. However, no theoretical justification had been given for its use and its extension to composite functions [40]. Gotovos *et al.* [41] proposed a level set estimation algorithm, which utilizes GPs to model the target function and exploits its inferred confidence bounds to drive the selection process. They provided an information-theoretic bound on the number of measurements needed to achieve a certain accuracy, when the underlying function is sampled from a GP.

In many mobile sensing problems, it is not enough to identify only a few specific regions but estimate the entire spatial field accurately. It can be formulated as a path planning problem to observe a spatial field at a set of sampling locations, and then making inference about the unobserved locations [42]. Choosing and visiting the sample locations so that one can have an accurate prediction (point prediction and/or prediction interval) is of great importance in soil science, agriculture, and air pollution monitoring [28]. The objective functions used are usually submodular, and thus, exhibit a diminishing returns property. Submodularity arises since nearby measurement locations are correlated [43]. Chekuri and Pal [44] introduced a quasi-polynomial time algorithm for maximizing a submodular objective along the path using a *recursive greedy strategy*. This algorithm was further extended by Binney *et al.* [45] for spatiotemporal fields using average variance reduction [46] as the objective function.

Zhang and Sukhatme [47] proposed an adaptive sampling algorithm consisting of a set of static nodes and a mobile robot tasked to reconstruct a scalar field. They assumed that the mobile robot can communicate with all the static nodes and acquire sensor readings from them. Based on this information, a path planner generates a path such that the resulting integrated MSE is minimized subject to the constraint that the robot has a finite amount of energy.

An important issue in designing robot paths is deciding the next measurement location [6], [48]–[50], often referred to as the exploration strategy. Traditionally, conventional sampling methods [51], such as raster scanning, simple random sampling, and stratified random sampling, have been used for single-robot exploration. Low *et al.* [52] presented an adaptive exploration strategy called adaptive cluster sampling. It was demonstrated to reduce mission time and yield more information about the environment. Their strategy performs better than a baseline sampling scheme called systematic sampling [53] using root mean squared error as a metric. A different adaptive multirobot exploration strategy called MASP was presented in [54] which performs both wide-area coverage and hotspot sampling using nonmyopic path planning. MASP allows for varying adaptivity and its performance is theoretically analyzed. Further, it was demonstrated to sample efficiently from a GP and *logGP*. However, the time complexity of implementing it depends on the map resolution, which limits its large-scale use. To alleviate this

computational difficulty, an information-theoretic approach was presented in [55]. The time complexity of the new approach was independent of the map resolution and less sensitive to the increasing robot team size. Garnett *et al.* [56] considered the problem of active search, which is also about sequential sampling from a domain of two (or more) classes. Their goal was to sample as many points as possible from one of the classes.

Yilmaz *et al.* [57] solved the adaptive sampling problem using mixed integer linear programming. Popa *et al.* [49] posed the adaptive sampling problem as a sensor fusion problem within the extended Kalman filter framework. Hollinger and Sukhatme [8] proposed a sampling-based motion planning algorithm that generates maximal informative trajectories for the mobile robots to observe their environment. Their information gathering algorithm extends ideas from rapidly-exploring random graphs. Using branch and bound techniques, they achieve efficient optimization of information gathering while also allowing for operation in continuous space with motion constraints. Cao *et al.* [58] presented two approaches to solve IPP for *in situ* active sensing of GP-based anisotropic spatial fields. Their proposed algorithms can tradeoff active sensing performance with computational efficiency. Ling *et al.* [9] proposed a nonmyopic adaptive GP planning framework endowed with a general class of Lipschitz continuous reward functions. Their framework can unify some active learning/sensing and Bayesian optimization criteria and offer practitioners flexibility to specify choices for defining new tasks. Tan *et al.* [59] introduced the receding-horizon cross-entropy trajectory optimization. Their focus was to sample around regions that exhibit extreme sensory measurements and much higher spatial variability, denoted as the region of interest. They used GP-UCB [60] as the optimization criteria which helps in exploring initially and converging on regions of interest eventually.

A naive implementation of GP prediction scales poorly with increasing training dataset size. Sparse GP frameworks can overcome this problem by using only a subset of the data to provide accurate estimates. A state-of-the-art sparse GP variant is SPGP [61]–[64]. The SPGP framework learns a pseudosubset that best summarizes the training data. Mishra *et al.* [65] introduced an online IPP framework AdaPP which uses SPGP.

C. Sensing With Multiple Robots

Mobile sensing can be made faster by distributing the task among several robots. Multirobot systems can do complex tasks and have been widely used in environmental sampling [66], coverage [67]. Robots can use local communication or control laws to achieve some collective goals.

Singh *et al.* [6] proposed a sequential allocation strategy that uses GP regression, which can be used to extend any single robot planning algorithm for the multirobot problem. Their procedure approximately generalizes any guarantees for the single-robot problem to the multirobot case. However, the approach works only when MI is the optimization objective. Cao *et al.* [58] presented two approaches along with their complexity analysis addressing a tradeoff between active sensing performance and time efficiency. Luo and Sycara [68] combined adaptive

sampling with information-theoretic criterion into the coverage control framework for model learning and simultaneous locational optimization. They presented an algorithm allowing for collaboratively learning the generalized model of density function using a mixture of GPs with hyperparameters learned locally from each robot. Kemna *et al.* [69] created a decentralized coordination approach which first splits the environments into Voronoi partitions and makes each vehicle then run within their own partition. Other multirobot approaches used in other domains, e.g., exploration and estimation with ground vehicles, include auction-based methods [70]–[72] and spatial segregation, typically through Voronoi partitioning [73], [74].

Tokekar *et al.* [10] presented a constant factor approximation algorithm for the case of accurately classifying each point in a spatial field. The first step in the algorithm is to determine potentially misclassified points and then to find a tour visiting neighborhoods of each potentially misclassified point. In this article, we study a regression version of the problem where every point is of interest. We exploit the properties of GP and squared-exponential kernel to find a constant-factor approximation algorithm. Before the details of the algorithms, we review some relevant background and useful properties of GPs and MSE.

D. Gaussian Processes

In GP regression, the posterior variance at any test location x is given by

$$\hat{\sigma}_{x|X}^2 = k(x, x) - \mathbf{k}(x, X) [\mathbf{K}(X, X) + \omega^2 \mathbf{I}]^{-1} \mathbf{k}(X, x) \quad (1)$$

where $\mathbf{K}(X, X)$ is the kernel matrix with entries, $\mathbf{K}_{pq} = k(x_p, x_q) = \sigma_0^2 \exp(-\frac{\|x_p - x_q\|^2}{2l^2})$. Here, σ_0^2 , l , ω^2 are signal variance, length scale, and additive independent and identically distributed Gaussian measurement noise, respectively [11]. We use the same value of length scale along each input dimension. Note that the posterior variance at a particular location x conditioned on set of observations at locations $X = \{x_1, \dots, x_n\}$ does not depend on the actual observation but only on the locations from where the observations are collected. Multiple observations at a location is equivalent to that location being counted as many times as the number of measurements. The kernel is a function that measures the *similarity* between two measurement locations [11].

Since the posterior variance is a function of only the measurement locations, the posterior variance for all points in the environment can be computed *a priori*, if the measurement locations are known, even without making any observations. In many implementations [13], [33], [75], the hyperparameters for the kernel k are tuned online as more data is gathered. As such, the hyperparameters may change with the observed data and the posterior variance will depend on the data observed, which may require adaptive planning. We assume that the hyperparameters are estimated *a priori*. This is done using prior data from the same or similar environments or a pilot deployment over a smaller region, as described in [5], [76], and [77]. Example applications are underwater inspection [76] and occupancy map

building [77], where prior data are used for determining hyperparameters before the actual deployment. Nevertheless, one can perform sensitivity analysis of the presented algorithms by varying the hyperparameters [78], [79].

The posterior mean $\hat{\mu}_{x|X}$ at a location x is given by a weighted linear combination of the observed data

$$\hat{\mu}_{x|X} = \mathbf{k}(x, X) [\mathbf{K}(X, X) + \omega^2 \mathbf{I}]^{-1} \mathbf{y} \quad (2)$$

where $\mathbf{y} = \{y_1, \dots, y_n\}$ denotes the observations at locations $X = \{x_1, \dots, x_n\}$.

E. Mean Square Error

MSE measures the expected squared difference between an estimator and the parameter the estimator is designed to estimate [80]. The MSE at a location x for an estimator \hat{f} is

$$MSE(\hat{f}(x)) = Var(\hat{f}(x)) + (E[\hat{f}(x) - f(x)])^2 \quad (3)$$

where (3) is the bias-variance expression for the estimator. GP predicted value $\hat{f}(x)$ at a location x is an unbiased estimator of the true value $f(x)$ [15] and has a normal distribution with mean given by (2), and variance given by (1). Among all linear and nonlinear estimators, GP is the best in terms of minimizing MSE [14], [81]. Further, GPs are unbiased and hence, the MSE at a location x is equal to the posterior variance of the predicted value, i.e.,

$$MSE(x) = \hat{\sigma}_{x|X}^2. \quad (4)$$

From (4), one can deduce that MSE for GPs is same as the posterior variance and hence, any guarantees for the posterior variance hold for MSE as well.

III. ALGORITHMS

In this section, we formally define the problems and the algorithms. We assume that the environment is a two-dimensional (2-D) area $U \subset \mathbb{R}^2$ and the underlying spatial field is an instance of a GP F [15]. F has an isotropic covariance function of the form

$$C_Z(x, x') = \sigma_0^2 \exp\left(-\frac{(x - x')^2}{2l^2}\right); \forall x, x' \in U \quad (5)$$

defined by a squared-exponential kernel where the hyperparameters σ_0^2 and l are known *a priori*. Let X denote the set of measurement locations within U produced by an algorithm.

Problem 1 (Placement): Find the minimum number of measurement locations, such that the MSE at each location in U is below $\Delta < \sigma_0^2$, i.e.,

$$\begin{aligned} &\text{minimize} && |X| \\ &\text{subject to} && MSE(x) \leq \Delta, \forall x \in U \end{aligned}$$

where $|X|$ is the cardinality of X and $MSE(x)$ is the MSE at location x .

Problem 2 (Mobile): Find the minimum time trajectory for a mobile robot that obtains a finite set of measurements at one or more locations in U , such that the MSE at each location in U

is less than Δ , i.e.,

$$\begin{aligned} &\text{minimize} && len(\tau) + \eta n(X) \\ &\text{subject to} && MSE(x) \leq \Delta \quad \forall x \in U. \end{aligned}$$

τ denotes the tour of the robot. Robot travels at unit speed, obtains one measurement in η units of time and obtains $n(X)$ total measurements.

The robot may be required to obtain multiple measurements from a single location. Therefore, the number of measurements $n(X)$ can be more than $|X|$. For multiple robots, their tours can start at the same starting location (often referred to as a *depot*) or can start at different locations. In this article, our focus is on the former case. The latter case is more appropriate when the robots must persistently monitor the environment.

Problem 3 (Multirobot): For k robots starting from a given starting location (depot), design a set of trajectories that collectively obtain a finite set of measurements at one or more locations in U , such that the MSE at each location in U is less than Δ , i.e.,

$$\begin{aligned} &\text{minimize} && \max_{i \in \{1, \dots, k\}} len(\tau_i) + \eta n(X_i) \\ &\text{subject to} && MSE(x) \leq \Delta \quad \forall x \in U \end{aligned}$$

where τ_i denotes the tour of the i th robot and X_i the subset of measurement locations covered by the i th robot. The robots travel at unit speed, obtain one measurement in η units of time. i th robot obtains $n(X_i)$ total measurements.

The solution for Problem 1 is a subset of the solution to Problem 2. Further, the solution for Problem 3 is derived from the solution for Problem 2. The three algorithms build on top of each other by: (1) finding a finite number of measurement locations for the robot; (2) finding a tour to visit all the measurement locations; and (3) splitting the tour from step 2 in multiple subtours for k robots. We exploit the properties of squared-exponential kernel to find the measurement locations. By knowing the value at a certain point within some tolerance, values at nearby points can be predicted albeit up to a larger tolerance.

A. Necessary and Sufficient Conditions

We start by deriving necessary conditions on how far a test location can be from its nearest measurement location. A test location corresponds to a point in the environment where we would like to make a prediction.

Lemma 1 (Necessary Condition): For any test location x , if the nearest measurement location is at a distance r_{\max} away, and

$$r_{\max} > l \sqrt{-\log\left(1 - \frac{\Delta}{\sigma_0^2}\right)} \quad (6)$$

then it is not possible to bring down the MSE below Δ at x .

Proof: Consider the posterior variance $\hat{\sigma}_x^2(n)$ at x [which is also equal to the MSE at x from (4)] after collecting n measurements, possibly from different locations. A lower bound on $\hat{\sigma}_x^2(n)$ can be obtained by assuming that all measurements were collected at the nearest location x_i to x . This is based on the fact that the closer the observation, lower the predictive variance. A mathematical proof for this is provided in Appendix.

Let the nearest measurement location x_i is distance r away from x . Assuming that all n measurements were collected at x_i , lower bound for posterior variance at x can be calculated using (1)

$$\hat{\sigma}_x^2(n) \geq \sigma_0^2 - \left[k(x, x_i), \dots, k(x, x_i) \right] \times \begin{bmatrix} \sigma_0^2 + \omega^2 & & \sigma_0^2 \\ & \ddots & \\ \sigma_0^2 & & \sigma_0^2 + \omega^2 \end{bmatrix}^{-1} \begin{bmatrix} k(x, x_i) \\ \vdots \\ k(x, x_i) \end{bmatrix}. \quad (7)$$

It is worth mentioning that the square matrix in (7) is of order $n \times n$ since there are n measurements. Substituting the value for $k(x, x_i) = \sigma_0^2 \exp(-\frac{r^2}{l^2})$ in (7) and performing the required matrix operations, we get

$$\hat{\sigma}_x^2(n) \geq \sigma_0^2 - \frac{\sigma_0^4}{\omega^2} \exp\left(\frac{-r^2}{l^2}\right) [1, \dots, 1] \times \begin{bmatrix} 1 - \frac{1}{n + \frac{\omega^2}{\sigma_0^2}} & & \frac{-1}{n + \frac{\omega^2}{\sigma_0^2}} \\ & \ddots & \\ \frac{-1}{n + \frac{\omega^2}{\sigma_0^2}} & & 1 - \frac{1}{n + \frac{\omega^2}{\sigma_0^2}} \end{bmatrix} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}. \quad (8)$$

Therefore

$$\hat{\sigma}_x^2(n) \geq \sigma_0^2 - \frac{\sigma_0^4}{\omega^2} \left(n \left(1 - \frac{1}{n + \frac{\omega^2}{\sigma_0^2}} \right) - \frac{n(n-1)}{n + \frac{\omega^2}{\sigma_0^2}} \right) \quad (9)$$

$$\geq \sigma_0^2 \left(1 - \frac{\exp\left(-\frac{r_{\max}^2}{l^2}\right)}{1 + \frac{\omega^2}{n\sigma_0^2}} \right). \quad (10)$$

Even if we had collected infinitely many measurements at the nearest location x_i , the posterior variance will still be lower bounded as

$$\hat{\sigma}_x^2(n) > \lim_{n \rightarrow \infty} \sigma_0^2 \left(1 - \frac{\exp\left(-\frac{r_{\max}^2}{l^2}\right)}{1 + \frac{\omega^2}{n\sigma_0^2}} \right) \quad (11)$$

$$= \sigma_0^2 \left(1 - \exp\left(-\frac{r_{\max}^2}{l^2}\right) \right). \quad (12)$$

If the posterior variance at x even with the infinitely many measurements collected at the nearest measurement location x_i [see (12)] is greater than Δ , i.e.,

$$\Delta < \sigma_0^2 \left(1 - \exp\left(-\frac{r_{\max}^2}{l^2}\right) \right) \Rightarrow r_{\max} > l \sqrt{-\log\left(1 - \frac{\Delta}{\sigma_0^2}\right)} \quad (13)$$

then it is not possible to bring down the MSE at x below Δ in any circumstance. ■

Next, we prove a sufficient condition that if every point in the environment where no measurement is obtained (*test* location) is sufficiently close to a measurement location, then we can make accurate predictions at each point.

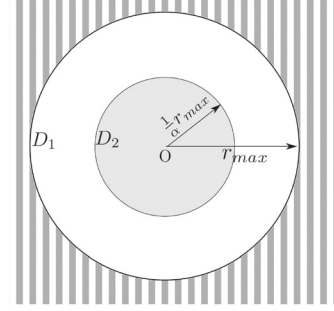


Fig. 2. Collecting n_α measurements at O suffices to make accurate predictions at all points inside disk D_2 (sufficient condition). No number of measurements at O can ensure predictive accuracy on points outside disk D_1 (necessary condition).

Lemma 2 (Sufficient Condition): For a test location $x \in U$, if there exists a measurement location $x_i \in X$, r distance away from x with n_{suff} measurements at x_i , such that

$$r \leq l \sqrt{-\log\left(\left(1 + \frac{\omega^2}{n_{\text{suff}}\sigma_0^2}\right)\left(1 - \frac{\Delta}{\sigma_0^2}\right)\right)} \quad (14)$$

then GP predictions at x will be accurate, i.e., MSE at x will be smaller than Δ .

Proof: From (10), we have an expression for variance of the posterior predictive distribution at x . Taking the other measurement locations into consideration can not increase the posterior variance at x . Information never hurts [82]! To prove the sufficiency, we consider n_{suff} measurements at x_i only and discard others knowing that the other locations can not increase the posterior variance at x . Bounding the expression in (10) with Δ results in

$$\Delta \geq \sigma_0^2 \left(1 - \frac{\exp\left(-\frac{r^2}{l^2}\right)}{1 + \frac{\omega^2}{n_{\text{suff}}\sigma_0^2}} \right) \quad (15)$$

$$\exp\left(-\frac{r^2}{l^2}\right) \geq \left(1 + \frac{\omega^2}{n_{\text{suff}}\sigma_0^2}\right) \left(1 - \frac{\Delta}{\sigma_0^2}\right) \quad (16)$$

$$r \leq l \sqrt{-\log\left(\left(1 + \frac{\omega^2}{n_{\text{suff}}\sigma_0^2}\right)\left(1 - \frac{\Delta}{\sigma_0^2}\right)\right)}. \quad (17)$$

Lemma 2 gives a sufficient condition for GP predictions to be accurate at any given test location $x \in U$. The following lemma shows that a finite number of measurements $n_{\text{suff}} = n_\alpha$, are sufficient to ensure predictive accuracy in a smaller disk of radius $\frac{1}{\alpha} r_{\max}$ around x_i , where $\alpha > 1$ (see Fig. 2).

Lemma 3: Given a disk of radius $\frac{1}{\alpha} r_{\max}$ centered at x_i , n_α measurements at x_i suffice to make accurate predictions for all points inside the disk, where

$$n_\alpha \geq \left\lceil \frac{\omega^2}{\sigma_0^2} \frac{1}{\left(1 - \frac{\Delta}{\sigma_0^2}\right)^{\frac{1}{\alpha^2-1}} - 1} \right\rceil. \quad (18)$$

Algorithm 1: DISKCOVER.

-
- 1: **Procedure**
 - 2: **Input:** An environment.
 - 3: **Output:** Measurement locations.
 - 4: **begin**
 - 1) Design a set \mathcal{X} of disks of radii r_{\max} which covers the environment and calculate a Maximal Independent Set (MIS) \mathcal{I} of \mathcal{X} greedily *i.e.*, $\mathcal{I} = \text{MIS}(\mathcal{X})$.
 - 2) Place disks of radii $3r_{\max}$ concentric with disks in \mathcal{I} . Let the set of $3r_{\max}$ radii disks is $\bar{\mathcal{X}}$.
 - 3) Cover each disk in $\bar{\mathcal{X}}$ with disks of radii $\frac{1}{\alpha}r_{\max}$ as shown in Fig. 3 and label centers of all disks of radii $\frac{1}{\alpha}r_{\max}$.
 - 4) Return all the labeled points in previous step as measurement locations.
-

end procedure

Proof: We want a sufficiency condition on the number of measurements n_α inside a disk of radius $\frac{1}{\alpha}r_{\max}$. Lemma 2 gives an upper bound on the radius of a disk such that all points inside the disk will be accurately predicted after n_α measurements at the center. We construct a disk (D_2 in Fig. 2), whose radius is equal to $\frac{1}{\alpha}r_{\max}$ such that

$$\frac{1}{\alpha}r_{\max} \leq l\sqrt{-\log\left(\left(1 + \frac{\omega^2}{n_\alpha\sigma_0^2}\right)\left(1 - \frac{\Delta}{\sigma_0^2}\right)\right)}. \quad (19)$$

Plugging in the value of r_{\max} from Lemma 1, squaring both sides in (19) and rearranging for n_α gives the required bound stated in Lemma 3. Ceiling function in (18) accounts for the fact that n_α is an integer. ■

A packing of disks of radius r_{\max} gives a lower bound on the number of measurements required to ensure predictive accuracy. On the other hand, a covering of disks of radius $\frac{1}{\alpha}r_{\max}$ gives us an upper bound on the number of measurements required. To solve Problem 1, what remains is to relate the upper and lower bound and present an algorithm to place the disks of radii $\frac{1}{\alpha}r_{\max}$.

B. Placement of Sensors for Problem 1

We use an algorithm similar to the one presented by Tekdas and Isler [83] for stationary sensor placement in order to track a target using bearing sensors. In their case, the goal is to place sensors such that irrespective of where the target is in the environment, there are at least three sensors forming a triangle that get good quality bearing information of the target. The show how to cover the environment with disks and place a triangle of sensors within each disk. The setup is different from the one we have; however, we use a similar disk coverage strategy as a subroutine here. The exact procedure is outlined in Algorithm 1.

Theorem 1: DISKCOVER (Algorithm 1) gives an $18\alpha^2$ -approximation for Problem 1 in polynomial time.

Proof: Denote the set of measurement locations computed by the optimal algorithm to solve the Problem 1 by X^* . The function MIS in Step 1 of Algorithm 1 computes a maximally

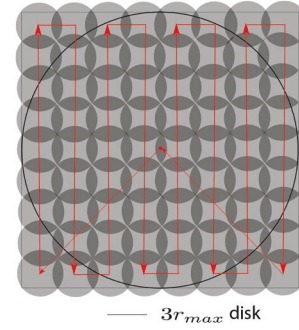


Fig. 3. We cover each $3r_{\max}$ radius disk with $\frac{1}{\alpha}r_{\max}$ radii disks (smaller gray disks) in lawn-mower pattern. $18\alpha^2$ disks suffice to cover the bigger disk. The locations of disks of radii $\frac{1}{\alpha}r_{\max}$ inside a disk of radius $3r_{\max}$ are obtained by covering the square circumscribing bigger disk with smaller squares inscribed in smaller disks. The centers of smaller squares coincide with the centers of smaller disks.

independent set of disks: the disks in \mathcal{I} are mutually nonintersecting (independent) and every disk in $\mathcal{X} \setminus \mathcal{I}$ intersects at least one disk in \mathcal{I} (maximal). The set \mathcal{I} can be computed by a simple polynomial greedy procedure: choose an arbitrary disk d from \mathcal{X} , add it to \mathcal{I} , remove all disks in \mathcal{X} which intersect d , and repeat the procedure until no such d exists.

An optimal algorithm will collect measurements from at least as many measurement locations as the cardinality of \mathcal{I} . This can be proved by contradiction. Suppose an algorithm visits measurement locations fewer than the number of disks in \mathcal{I} . In that case, there will exist at least one disk of radius r_{\max} in \mathcal{I} which will not contain a measurement location. This means that there will be at least a point in that disk which will be more than r_{\max} away from each measurement location. From Lemma 1, the robot can never make accurate predictions at that point and hence violating the constraint in Problem 1. Hence

$$|\mathcal{I}| \leq |X^*|. \quad (20)$$

Every disk in \mathcal{X} intersects at least one disk in \mathcal{I} and hence, lies within $3r_{\max}$ of the center of a disk in \mathcal{I} . As a result, $\bar{\mathcal{X}}$ disks cover all the \mathcal{X} disks and hence, the entire environment.³

Collecting measurements from $18\alpha^2$ locations inside a $3r_{\max}$ disk suffice to make accurate predictions in that disk (satisfying the Problem 1 constraint for points belonging to that disk) as illustrated in Fig. 3. DISKCOVER collects measurement from $18\alpha^2$ such locations per disk in $\bar{\mathcal{X}}$. It collects measurements from a total of $18\alpha^2|\bar{\mathcal{X}}|$ locations, hence, satisfying the constraint for all points in the area covered by union of $\bar{\mathcal{X}}$ disks. Since, union of $\bar{\mathcal{X}}$ disks covers the entire environment, DISKCOVER satisfies the constraint for all points in the environment. Multiplying both sides of (20) with $18\alpha^2$, we get, $18\alpha^2|\mathcal{I}| \leq 18\alpha^2|X^*|$. Note that $|\bar{\mathcal{X}}| = |\mathcal{I}|$. Hence

$$18\alpha^2|\bar{\mathcal{X}}| \leq 18\alpha^2|X^*| \quad (21)$$

$$n_{\text{DISKCOVER}} \leq 18\alpha^2|X^*| \quad (22)$$

³Note that $3r_{\max}$ is the minimum radius of the bigger disks to guarantee that the entire environment is always covered. In specific instances, it may be possible to cover the environment with smaller than $3r_{\max}$ by selecting the MIS using a well-designed heuristic. However, there are environments where $3r_{\max}$ will be necessary.

Algorithm 2: DISKCOVERTOUR.

-
- 1: **Procedure**
 - 2: **Input:** A set of measurement locations calculated from Algorithm 1.
 - 3: **Output:** An approximate optimal tour visiting all the measurement locations.
 - 4: **begin**
 - 1) Calculate approximate TSP tour visiting centers of the $3r_{\max}$ radius disks (set $\bar{\mathcal{X}}$) disks.
 - 2) Cover $\bar{\mathcal{X}}$ disk containing the starting location in lawn-mower pattern visiting the centers of corresponding disks of radius $\frac{1}{\alpha}r_{\max}$ and make n_α measurements at each center point.
 - 3) Move to the center of next $\bar{\mathcal{X}}$ disk along the tour calculated in Step 1.
 - 4) Repeat Steps 2 and 3 until all $\bar{\mathcal{X}}$ disks are covered.
- end procedure**
-

where $n_{\text{DISKCOVER}}$ is the number of measurement locations for DISKCOVER. ■

C. Finding an Approximate Optimal Trajectory for Problem 2

The algorithm for Problem 2 builds on the algorithm presented in the previous section. The locations where measurements are to be made become the locations that are to be visited by the robot. The robot must obtain at least n_α measurements at the center of each disk of radius $\frac{1}{\alpha}r_{\max}$. A pseudocode of the algorithm is presented in the Algorithm 2.

Theorem 2: DISKCOVERTOUR (Algorithm 2) yields a constant-factor approximation algorithm for Problem 2 in polynomial time.

Proof: From Theorem 1, we have a constant approximation bound on number of measurement locations. Let the time (travel and measurement time) taken by the optimal algorithm be T_1^* . Using notation from Theorem 1, we assume that the optimal traveling salesperson with neighborhoods (TSPN) time to visit disks in \mathcal{I} be $T_{\mathcal{I}}^*$. In TSPN, we are given a set of geometric neighborhoods, and the objective is to find the shortest tour that visits at least one point in each neighborhood (disks in this case) [10]. The optimal algorithm will visit at least all disks once in \mathcal{I} which gives the following minimum bounds on the optimal travel time (T_{travel}^*) and optimal measurement time (T_{measure}^*)

$$T_{\mathcal{I}}^* \leq T_{\text{travel}}^*; \eta|\mathcal{I}| \leq T_{\text{measure}}^*. \quad (23)$$

Let the optimal time to visit the centers of disks in \mathcal{I} be $T_{\mathcal{I}C}^*$. An upper bound on $T_{\mathcal{I}C}^*$ can be established by the fact that upon visiting each disk, the robot can visit the center of that disk and return back by adding an extra tour length of $2r_{\max}$, i.e., a detour of maximum length $|\mathcal{I}| \times 2r_{\max}$ for all disks in \mathcal{I} . As a result: $T_{\mathcal{I}C}^* \leq T_{\mathcal{I}}^* + 2r_{\max}|\mathcal{I}|$. Using inequality from (23): $T_{\mathcal{I}C}^* \leq T_{\text{travel}}^* + 2r_{\max}|\mathcal{I}|$. For any disk in $\bar{\mathcal{X}}$, the length of lawn-mower path starting from the its center and return back (see Fig. 3) after visiting all center points of $\frac{1}{\alpha}r_{\max}$ disks will be of order $\mathcal{O}(\alpha^2)r_{\max}$. Hence, the total travel time

for DISKCOVERTOUR is: $T_C + |\mathcal{I}|\mathcal{O}(\alpha^2)r_{\max}$, where T_C is the $(1 + \epsilon)$ -approximated time with respect to the optimal TSP tour returned by the $(1 + \epsilon)$ -approximation algorithm to visit the centers of the disks in $\bar{\mathcal{X}}$ (or \mathcal{I} disks since they are concentric). T_C can be calculated in polynomial time [84] having bounds: $T_C \leq (1 + \epsilon)T_{\mathcal{I}C}^*$, with $T_{\mathcal{I}C}^*$ being the optimal TSP time to visit the centers of $\bar{\mathcal{X}}$ disks. Measurement time for DISKCOVERTOUR is $18\alpha^2\eta n_2|\mathcal{I}|$. Hence, the total time T_{alg}^1 for DISKCOVERTOUR is

$$T_{\text{alg}}^1 = T_C + |\mathcal{I}|\mathcal{O}(\alpha^2)r_{\max} + 18\alpha^2\eta n_2|\mathcal{I}| \quad (24)$$

$$\leq (1 + \epsilon)T_{\mathcal{I}C}^* + \mathcal{O}(\alpha^2)r_{\max}|\mathcal{I}| + 18\alpha^2\eta n_2|\mathcal{I}| \quad (25)$$

$$\leq (1 + \epsilon)(T_{\text{travel}}^* + 2r_{\max}|\mathcal{I}|) + \mathcal{O}(\alpha^2)r_{\max}|\mathcal{I}| + 18\alpha^2\eta n_2|\mathcal{I}| \quad (26)$$

where n_2 is the number of sufficient measurements required inside a disk of radius $\frac{1}{2}r_{\max}$ (Lemma 3 with $\alpha = 2$). Length of any tour that visits k nonoverlapping equal size disks of radii r is at least $0.24kr$ [85], which gives $0.24r_{\max}|\mathcal{I}| \leq T_{\mathcal{I}}^*$. Combining this result with (23) modifies the bounds in (26) as

$$T_{\text{alg}}^1 \leq \left((1 + \epsilon) \left(1 + \frac{2}{.24} \right) + \frac{\mathcal{O}(\alpha^2)}{.24} \right) T_{\text{travel}}^* + 18\alpha^2 n_2 T_{\text{measure}}^* \quad (27)$$

$$\leq \max \left(9.33(1 + \epsilon) + \frac{82}{.24}, 72n_2 \right) (T_{\text{travel}}^* + T_{\text{measure}}^*), \quad (28)$$

$$\leq \max \left(9.33(1 + \epsilon) + \frac{\mathcal{O}(\alpha^2)}{.24}, 18\alpha^2 n_2 \right) T_1^* \quad (29)$$

$$\leq cT_1^* \quad (30)$$

where c , a constant, is larger one of the two quantities inside the bracket in (30). ■

Note that the Algorithm 2 collects same number of measurements from each measurement location. There may be another algorithm that collects different number of measurements from different locations which may result in better performance. This modification is an avenue for future work.

D. Finding an Approximate Optimal Trajectory for Problem 3

When one robot can not handle a large territory, to speed up the task, k robots can be sent to collectively visit all measurement locations. A natural objective is to ensure that no robot has too large of a task. Hence, we choose our optimization criterion as minimizing the maximum of the k -robot tour costs. This is equivalent to minimizing the time taken by the last robot to return back to the common starting location. Our proposed algorithm only works if the robots start and return back to the same location called depot. Any measurement location can be chosen as the depot but in our case, we assume that the robots start from and return back to a predefined depot.

We now describe an algorithm which employs a tour-splitting heuristic to plan for k robots. We modify the heuristic proposed

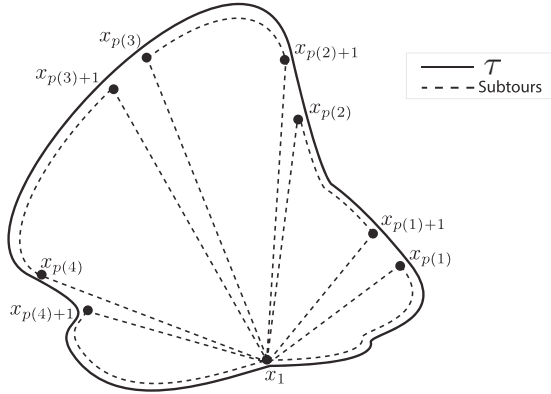


Fig. 4. Splitting the tour for one robot (τ) into 5 subtours. The solid line shows an initial single robot tour τ starting and ending at x_1 . The dotted lines denote the individual robot subtours starting and ending at x_1 obtained by splitting the single tour τ .

Algorithm 3: k -DISKCOVERTOUR.

- 1: **Procedure**
- 2: **Input:** Tour calculated from Algorithm 2, Depot location x_1 .
- 3: **Output:** k approximate optimal paths visiting all measurement locations collectively.
- 4: **begin**
 - 1) For j^{th} robot, $1 \leq j < k$, find the last measurement location $x_{p(j)}$ such that the time taken to travel from x_1 to $x_{p(j)}$ along τ is not greater than $\frac{j}{k}(T_{\text{alg}}^1 - (2l_{\text{max}} + \eta n_2)) + (l_{\text{max}} + \eta n_2)$.
 - 2) Obtain k subtours as $R_1 = (x_1, \dots, x_{p(1)}, x_1)$,
 $R_2 = (x_1, x_{p(1)+1}, \dots, x_{p(2)}, x_1), \dots$
 $R_k = (x_1, x_{p(k-1)+1}, \dots, x_n, x_1)$.

end procedure

by Frederickson *et al.* [86] to account for the measurement time, and not just the travel time.

Let the output tour of the robot from Algorithm 2 be denoted by τ and l_{max} be the distance of farthest measurement location from the depot.

Theorem 3: k -DISKCOVERTOUR (Algorithm 3) yields a $(c + 2)$ approximation algorithm for Problem 3 in polynomial time, given a c -approximation algorithm for Problem 2.

Proof: First, we prove that the time taken along every subtour is bounded and eventually show that the bound is within a constant factor of the optimal time. With k robots, let the subtours for first and k th robot are $x_1 \rightarrow x_{p(1)} \rightarrow x_1$ and $x_1 \rightarrow x_{p(k-1)+1} \rightarrow x_1$, respectively (an example with $k = 5$ is shown in Fig. 4). Subtours for the remaining robots can be denoted by $x_1 \rightarrow x_{p(j-1)+1} \rightarrow x_{p(j)} \rightarrow x_1$, where $1 < j < k$.

Substituting $j = 1$ in Algorithm 3, the time to travel from x_1 to $x_{p(1)}$ along τ , $T(x_1 \xrightarrow{\tau} x_{p(1)})$ is no greater than $\frac{1}{k}(T_{\text{alg}}^1 - (2l_{\text{max}} + \eta n_2)) + (l_{\text{max}} + \eta n_2)$. Time for the first subtour is hence bounded by $T(x_1 \xrightarrow{\tau} x_{p(1)}) + T(x_{p(1)} \rightarrow x_1)$, *i.e.*, $\frac{1}{k}(T_{\text{alg}}^1 - (2l_{\text{max}} + \eta n_2)) + (l_{\text{max}} + \eta n_2) + l_{\text{max}}$. From

the condition in Algorithm 3, we know that $x_{p(k-1)}$ is the last location such that

$$T(x_1 \xrightarrow{\tau} x_{p(k-1)}) \leq \frac{k-1}{k}(T_{\text{alg}}^1 - (2l_{\text{max}} + \eta n_2)) + (l_{\text{max}} + \eta n_2) \quad (31)$$

and hence

$$T(x_1 \xrightarrow{\tau} x_{p(k-1)+1}) \geq \frac{k-1}{k}(T_{\text{alg}}^1 - (2l_{\text{max}} + \eta n_2)) + (l_{\text{max}} + \eta n_2). \quad (32)$$

Subtracting both sides from T_{alg}^1

$$T_{\text{alg}}^1 - T(x_1 \xrightarrow{\tau} x_{p(k-1)+1}) \leq T_{\text{alg}}^1 - \frac{k-1}{k}(T_{\text{alg}}^1 - (2l_{\text{max}} + \eta n_2)) + (l_{\text{max}} + \eta n_2) \quad (33)$$

which gives

$$T(x_{p(k-1)+1} \xrightarrow{\tau} x_1) \leq \frac{1}{k}(T_{\text{alg}}^1 - (2l_{\text{max}} + \eta n_2)) + (3l_{\text{max}} + 2\eta n_2), \quad (34)$$

and hence, time for the last subtour is bounded by $T(x_1 \rightarrow x_{p(k-1)+1}) + T(x_{p(k-1)+1} \xrightarrow{\tau} x_1)$, *i.e.*, $\frac{1}{k}(T_{\text{alg}}^1 - (2l_{\text{max}} + \eta n_2)) + (3l_{\text{max}} + 2\eta n_2) + l_{\text{max}}$. Similar inequalities can be derived for remaining subtours as follows. For $1 \leq j \leq k-2$, following inequalities hold from Algorithm 3:

$$T(x_1 \xrightarrow{\tau} x_{p(j)+1}) \geq \frac{j}{k}(T_{\text{alg}}^1 - (2l_{\text{max}} + \eta n_2)) + (l_{\text{max}} + \eta n_2), \quad (35)$$

$$T(x_1 \xrightarrow{\tau} x_{p(j+1)}) \leq \frac{j+1}{k}(T_{\text{alg}}^1 - (2l_{\text{max}} + \eta n_2)) + (l_{\text{max}} + \eta n_2). \quad (36)$$

Subtracting (35) from (36) results in

$$T(x_{p(j)+1} \xrightarrow{\tau} x_{p(j+1)}) \leq \frac{1}{k}(T_{\text{alg}}^1 - (2l_{\text{max}} + \eta n_2)) \quad (37)$$

i.e., the time taken along remaining subtours, $T(x_1 \rightarrow x_{p(j-1)+1} \xrightarrow{\tau} x_{p(j)} \rightarrow x_1)$, where $1 < j < k$, is also bounded by $\frac{1}{k}(T_{\text{alg}}^1 - (2l_{\text{max}} + \eta n_2)) + 2l_{\text{max}}$. Hence, we can conclude that the time taken along each subtour does not exceed $\frac{1}{k}(T_{\text{alg}}^1 - (2l_{\text{max}} + \eta n_2)) + (4l_{\text{max}} + 2\eta n_2)$.

Let T_{alg}^k be the time taken for largest of the k subtours generated by the Algorithm 3, and T_k^* be the cost of the largest subtour in an optimal solution to Problem 3. We have

$$T_{\text{alg}}^k \leq \frac{1}{k}(T_{\text{alg}}^1 - (2l_{\text{max}} + \eta n_2)) + (4l_{\text{max}} + 2\eta n_2) \quad (38)$$

$$\leq \frac{T_{\text{alg}}^1}{k} + (2l_{\text{max}} + \eta n_2) \left(2 - \frac{1}{k}\right) \quad (39)$$

from the triangle inequality $T_k^* \geq \frac{1}{k}T_1^*$. It is natural to think that at least one robot will have to go to the farthest location from the depot and come back from there after collecting n_2

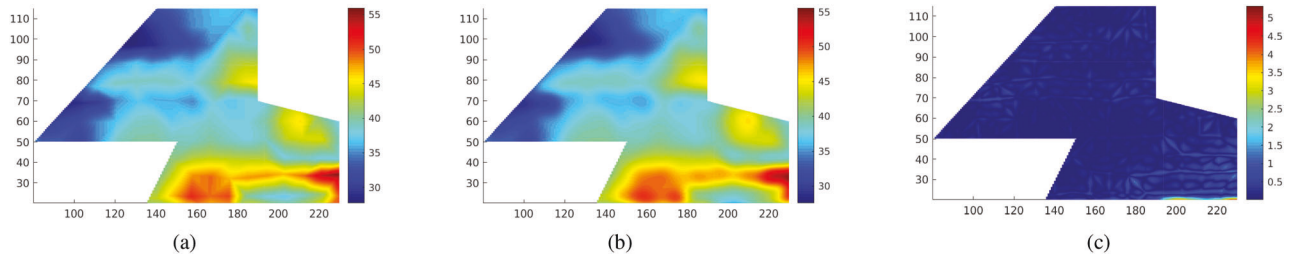


Fig. 5. Actual and predicted OM content comparison. The farm is shown as the colored region with the colorbar denoting concentrations at different locations. All distance units are in *meter*. (a) Actual OM content (ppm). (b) Predicted OM content (ppm). (c) Prediction error between the actual and the predicted OM content (ppm).

measurements which gives us a lower bound on the output of the optimal algorithm, i.e., $2l_{\max} + \eta m_2 \leq T_k^*$. Combining these results with (30), we get

$$T_{\text{alg}}^k \leq \frac{c}{k} T_1^* + T_k^* \left(2 - \frac{1}{k} \right) \quad (40)$$

$$\leq \left(c + 2 - \frac{1}{k} \right) T_k^* \quad (41)$$

$$\leq (c + 2) T_k^*. \quad (42)$$

■

IV. EMPIRICAL EVALUATION

In this section, we report results from empirical evaluation of the theoretical results. We show qualitative and quantitative comparison of our algorithms with other baseline strategies through simulations using precision agriculture as our motivating example.

Dataset: We use a real-world dataset [87], collected from a farm, consisting of organic matter (OM) measurements manually collected from several hundred locations within the farm. The maximum and minimum values of the underlying field are 54.6 *parts per million (ppm)* and 25.4 ppm, respectively shown by the colorbar [see Fig. 5(a)]. Taking this into account, we set Δ to be equal to 4 which is 10% of the average of maximum and minimum field values. We use a simulated sensor that returns a noisy version of the ground truth measurement with an additive Gaussian noise of variance $\omega = 0.0361$.

The squared-exponential kernel has three hyperparameters—length scale (l), signal variance (σ_0^2), and noise variance (ω^2). The values of l , σ_0 , and ω^2 were estimated to be 8.33 m, 12.87, and 0.0361, respectively by minimizing the negative log-marginal likelihood of the manually collected data. We assume that the estimated values are the true values of the kernel hyperparameters. In a general application where some prior data are available, the hyperparameters can be estimated in a similar way. We used the GPML toolbox to perform the necessary GP operations [88].

A. Qualitative Example

Stationary Sensor Placement: The final predicted OM content after performing inference using the measurements obtained is

shown in Fig. 5(b). This predicted OM content is the average of ten trials. In each trial, the reported value by the sensor can be different even at the same location because of the simulated noise. Fig. 5(c) shows a plot of the prediction error averaged over those ten trials. We observe that the average prediction error is below $\Delta = 4$ ppm at each location in the environment. It is important to mention that average prediction error is not same as the MSE. The MSE at a location is the expected squared error in prediction at that location. The average prediction error, referred as empirical MSE in the following text, is an empirical estimate of that expectation. As the number of runs increases, the empirical MSE will converge to the actual MSE. We verify it through simulations and report the results later. Our theoretical guarantees hold for the MSE and not for the empirical MSE. However, one can expect that the empirical MSE will also be less than the predefined threshold Δ given enough trials. The regions where the OM content changes sharply tend to be more erroneously predicted as shown by the lighter colored regions in Fig. 5(c). This can be attributed to the inherent smoothness assumptions of a squared-exponential kernel.

Single and Multirobot Tours: The measurement locations computed by the DISKCOVERTOUR are shown in Fig. 6(a). As postprocessing, we removed the redundant measurement locations in overlapping $3r_{\max}$ radii disks. After performing this step, the total number of measurement locations was 2320. A covering of the farm with disks of radii $3r_{\max}$ and an approximate optimal tour visit the centers of those disks calculated by DISKCOVERTOUR is shown in Fig. 6(b). We compute the optimal TSP tour since this is a reasonably sized instance. The lawn-mower detours visiting individual $3r_{\max}$ disks have been omitted to make the figure more legible. For the multirobot version, we assume that we have three robots. Splitting of a single robot tour [see Fig. 6(b)] in three sub-tours is shown in Fig. 6(c). The robots start from a common depot.

Varying Values of Δ : In some applications, one may be interested in having more accurate predictions in some parts of the environment than others. Our algorithms provides a way to choose locations and plan paths in such applications as well. To demonstrate this, we divide the farm in three subenvironments that have different Δ tolerances. The left-most, middle, and right-most regions have thresholds of $\Delta = 6, 4, \text{ and } 2$ ppm,

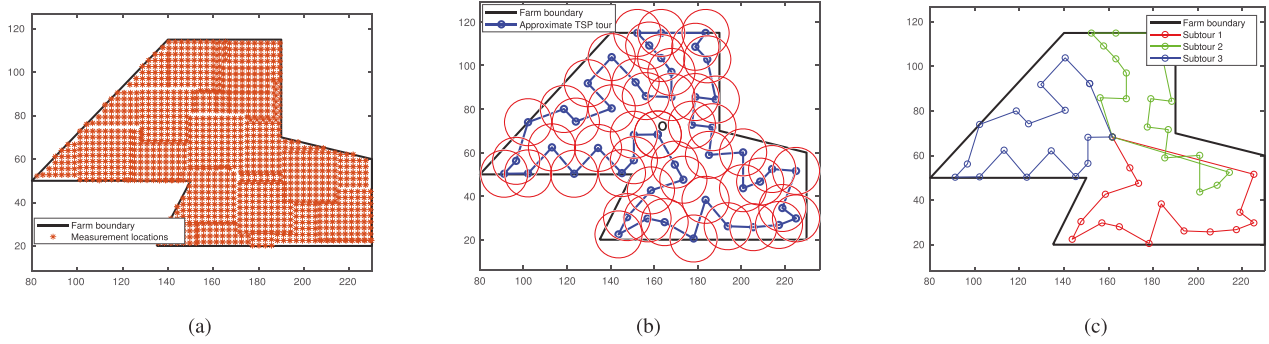


Fig. 6. Measurement locations and the tours computed by DISKCOVER and k -DISKCOVERTOUR. For Fig. 6(b) and (c), the complete tours that take detours to visit all the locations in Fig. 6(a) have been omitted to make the figures more legible. (a) Measurement locations calculated by the DISKCOVER algorithm. There are 1012 measurement locations. (b) Red disks are of radii $3r_{\max}$ which are concentric with disks of radii r_{\max} in \mathcal{I} . The depot is denoted by O. (c) All robots start from and return to the depot after making measurements.

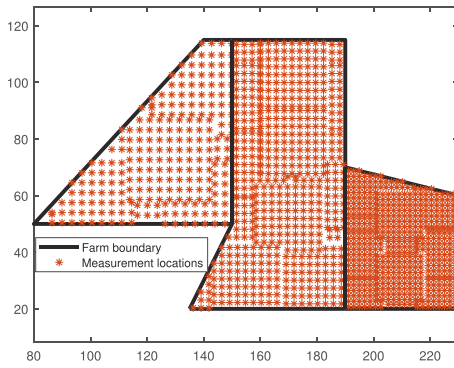


Fig. 7. Measurement locations for different values of Δ .

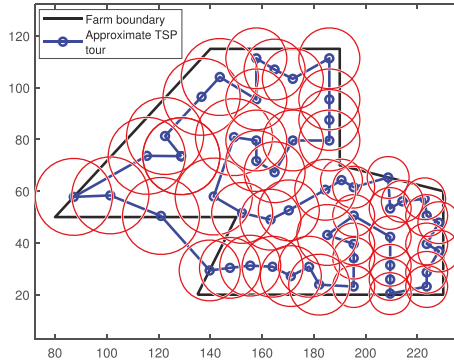


Fig. 8. Approximate TSP tour to visit the $3r_{\max}$ disks. r_{\max} values depend on Δ (13) and hence, vary in different Δ subregions. Note the shrinking size of disks as one moves toward right.

respectively. We solve for the measurement locations independently in each region. The corresponding r_{\max} values were calculated to be 4.97, 3.93, and 2.70 m, respectively using (13). Fig. 7 shows the measurement locations. One can qualitatively observe that the algorithm places fewer measurement locations in the left-most subenvironment which allows for the highest error tolerance.

An approximate TSP tour visiting the centers of all $3r_{\max}$ disks, in all three regions, is shown in Fig. 8. The size of the

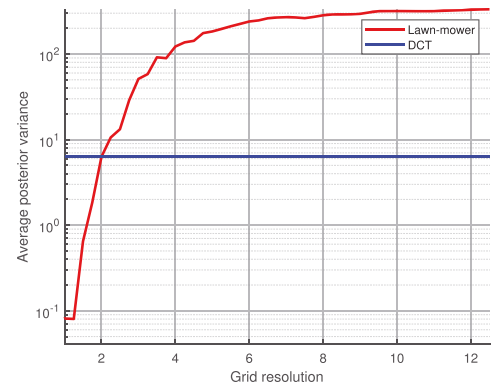


Fig. 9. Average posterior variance for varying degree of lawn-mower resolutions.

disks shrinks as one moves to the right-most subregion which has the least tolerance for prediction error. The TSP tour goes outside the environment in this case, which may be feasible if an aerial robot is used to monitor the farm. In case of applications, where the robot must stay inside the environment, we can enforce this constraint by replacing the Euclidean edge weights in the TSP input graph with the length of the shortest path between two vertices inside the environment.

B. Comparisons With Predefined Lawn-Mower Tours

One can observe from Fig. 6(a) that the measurement location pattern closely resembles a lawn-mower pattern. It motivated us to compare the performance of our algorithms and with lawn-mower plan. Figs. 9 and 10 show the average posterior variance and average empirical (for ten trials) MSE respectively for a predefined lawn-mower pattern with varying grid resolutions on a semilogarithmic scale. Note that the posterior variance at a test location is always same in each trial because it is not a function of the actual measurement value. The blue horizontal line corresponds to DCT and is shown for the sake of comparison.

A plot of the time taken by the robot to cover lawn-mower patterns with various grid resolutions is shown in Fig. 11. The

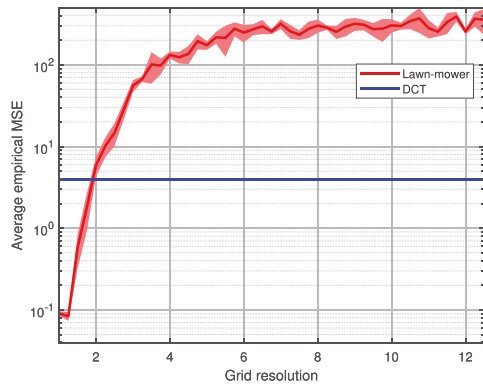


Fig. 10. Average empirical MSE for varying degree of lawn-mower resolutions.

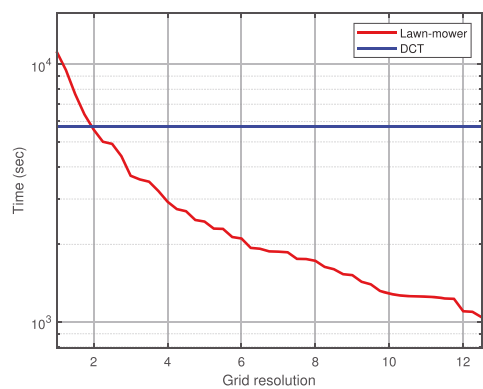


Fig. 11. Time spent by the robot with lawn-mower planners of different grid resolutions.

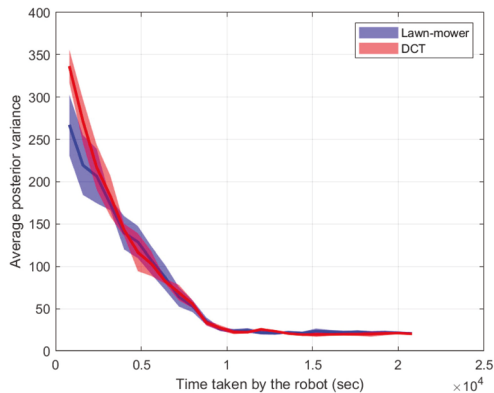


Fig. 12. Average posterior variance as a function of time spent by the robot.

lawn-mower lines in Figs. 9–11 intersect the DCT lines at approximately a resolution of 2 m. It suggests that one would need to create a grid of approximately that resolution to achieve same performance as DCT. Fig. 12 shows the average posterior variance for DCT and a predefined lawn-mower of resolution 2.4 m as a function time elapsed along a deployment (averaged over 10 deployments). We chose a resolution of 2.4 m since a lawn-mower planner with this resolution has approximately the

same number of measurement locations as DISKCOVERTOUR. We observe that both perform almost the same empirically.

One may wonder why we cannot simply use the lawn-mower pattern, instead of DISKCOVERTOUR. To create a lawn-mower pattern, one would need to pick a grid resolution. There is no systematic way of picking this resolution without enumerating a few combinations to analyze the tradeoff between time and posterior variance or MSE. This can be wasteful. Instead, we present a systematic way of planning the measurement locations and give explicit theoretical guarantees on time and MSE or variance.

C. Comparison With Other Baselines

A comparison between DISKCOVERTOUR and two baselines, entropy-based, and MI-based planner is shown in Fig. 13. The measurement locations for the entropy-based and MI-based planners were calculated greedily, i.e., picking the next location at the point of maximum entropy and MI, respectively as described in [5]. We study the average posterior variance and average empirical MSE in prediction as a function of the total time (measurement plus traveling) spent by the robot on the farm for each planner. After finding the measurement locations for each planner separately, TSP tours visiting those locations were calculated. The X -axis in Fig. 13 shows the time taken along a tour and the Y -axis shows the respective metrics based on measurements collected until that point in time along the tour (averaged over ten trials). We observe that DISKCOVERTOUR performs at par with other planners. The entropy-based planner results in the most significant reduction in posterior variance and average empirical MSE initially. This can be explained by the fact that the entropy-based planning tends to spread the measurement locations far from each other resulting in covering a bigger portion of the environment initially. However, DISKCOVERTOUR converges to a lower value of average empirical MSE and average posterior variance.

D. MSE and Variance

We verify our hypothesis that MSE is equal to the posterior variance for GPs. A plot of the mean percent difference between the empirical MSE and the posterior variance is shown in Fig. 14. The mean is computed over approximately 5600 test locations which are different from the measurement locations and placed on a grid. As the number of trials increases, the mean difference between empirical MSE, which is essentially the MSE given enough number of trials, and the posterior variance decreases implying that the empirical MSE converges to the posterior variance asymptotically. In each trial, the measurement locations, test locations, and the hyperparameters are same, and therefore the variance estimates are same as well. However, the predicted value in each trial, and hence the prediction error, may be different since the actual measurement collected can be different in each trial due to the simulated noise. The effect of noise will decrease as one computes empirical estimate over a larger number of trials.

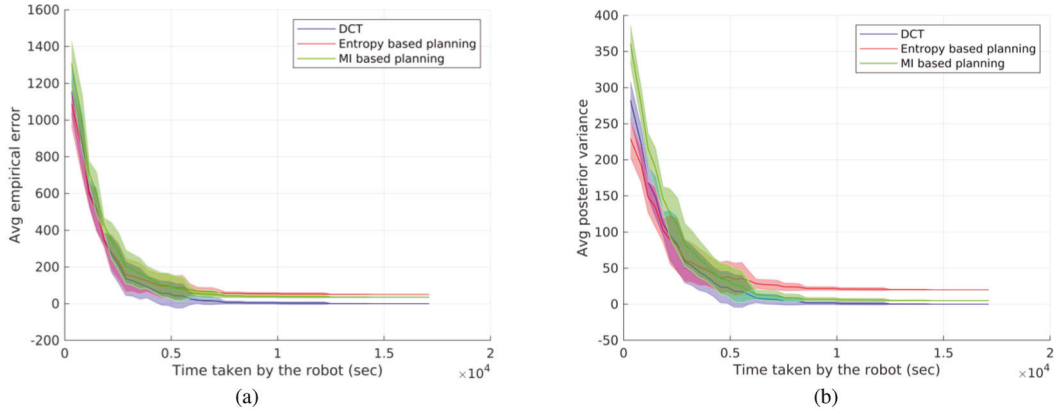


Fig. 13. DISKCOVERTOUR performs comparably with entropy-based and MI-based strategies. The shaded regions correspond to the standard deviation taken over ten trials. (a) Average empirical MSE. (b) Average posterior variance.

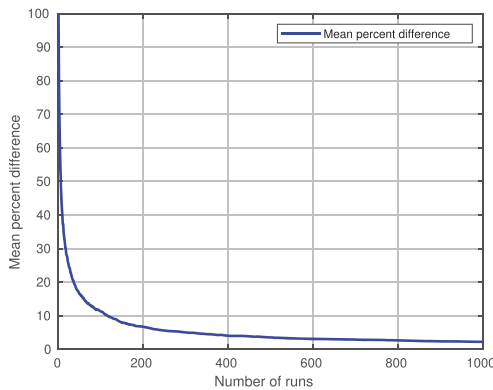


Fig. 14. Mean percentage difference between the empirical MSE and the posterior variance.

V. CONCLUSION

In this article, we studied several problems—placing the minimum number of stationary sensors to track a spatial field, mapping a spatial field by a single as well as multiple robots while minimizing the time taken by the robots. For all the problems, we proposed polynomial-time approximation algorithms to ensure that the MSE in prediction the underlying spatial field was smaller than a predefined threshold at each point. We also derived the lower bounds on the performance of any algorithm (including optimal) to solve respective problems are provided. We showed that it was possible to learn a given spatial field accurately with high confidence without planning adaptively. Note that, if the kernel parameters were optimized online, then, one would require an adaptive strategy.

The algorithms suggested in this article performed comparatively with the baseline planners developed earlier. Our algorithms had theoretical bounds on their performance. The algorithms could also be generalized to 3-D mapping, even though we illustrated using 2-D examples. The disks in the 2-D case would be replaced by spheres in 3-D. The disk packing/covering problem became a sphere packing/covering. The tour would need to visit points in 3-D, as opposed to 2-D. The existing TSP algorithms already apply to the 3-D case [42]. Our ongoing

work is on developing competitive strategies for spatio-temporal learning and deriving similar guarantees for adaptive cases.

APPENDIX

Proof: Consider two measurement locations x_1, x_2 and a test location x such that x_1 is closer to x . The posterior variance at x if a measurement was collected at x_1 can be computed as follows:

$$\hat{\sigma}_{x|x_1}^2 = k(x, x) - k(x, x_1)K(x_1, x_1)^{-1}k(x_1, x) \quad (43)$$

$$= \sigma_0^2 \left(1 - \exp\left(-\frac{\|x - x_1\|^2}{l^2}\right) \right). \quad (44)$$

Similarly, the posterior variance at x if a measurement was collected at x_2

$$\hat{\sigma}_{x|x_2}^2 = \sigma_0^2 \left(1 - \exp\left(-\frac{\|x - x_2\|^2}{l^2}\right) \right). \quad (45)$$

From $\|x - x_1\|^2 < \|x - x_2\|^2$ and $f(x) = -\exp(-x)$ being a monotonically increasing function, we have

$$-\exp\left(-\frac{\|x - x_1\|^2}{l^2}\right) < -\exp\left(-\frac{\|x - x_2\|^2}{l^2}\right). \quad (46)$$

Using this to compare (44) and (45) one can easily see that $\hat{\sigma}_{x|x_1}^2 < \hat{\sigma}_{x|x_2}^2$. ■

REFERENCES

- [1] W. Aktar, D. Sengupta, and A. Chowdhury, "Impact of pesticides use in agriculture: Their benefits and hazards," *Interdisciplinary Toxicology*, vol. 2, no. 1, pp. 1–12, 2009.
- [2] L. Adzighli and D. Yuewen, "Assessing the impact of oil spills on marine organisms," *J. Oceanogr. Mar. Res.*, vol. 6, no. 179, pp. 472–479, 2018.
- [3] J. Blachowski, "Spatial analysis of the mining and transport of rock minerals (aggregates) in the context of regional development," *Environ. Earth Sci.*, vol. 71, no. 3, pp. 1327–1338, Feb. 2014.
- [4] G. Werner-Allen *et al.*, "Deploying a wireless sensor network on an active volcano," *IEEE Internet Comput.*, vol. 10, no. 2, pp. 18–25, Mar. 2006.
- [5] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies," *J. Mach. Learn. Res.*, vol. 9, pp. 235–284, 2008.
- [6] A. Singh, A. Krause, C. Guestrin, and W. J. Kaiser, "Efficient informative sensing using multiple robots," *J. Artif. Intell. Res.*, vol. 34, pp. 707–755, 2009.

- [7] R. Ouyang, K. H. Low, J. Chen, and P. Jaillet, "Multi-robot active sensing of non-stationary Gaussian process-based environmental phenomena," in *Proc. Int. Conf. Auton. Agents Multi-Agent Syst.*, 2014, pp. 573–580.
- [8] G. A. Hollinger and G. S. Sukhatme, "Sampling-based robotic information gathering algorithms," *Int. J. Robot. Res.*, vol. 33, no. 9, pp. 1271–1287, 2014.
- [9] C. K. Ling, K. H. Low, and P. Jaillet, "Gaussian process planning with Lipschitz continuous reward functions: Towards unifying Bayesian optimization, active learning, and beyond," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 1860–1866.
- [10] P. Tokekar, J. Vander Hook, D. Mulla, and V. Isler, "Sensor planning for a symbiotic UAV and UGV system for precision agriculture," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1498–1511, Dec. 2016.
- [11] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006.
- [12] E. Schulz, M. Speekenbrink, and A. Krause, "A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions," *J. Math. Psychol.*, vol. 85, pp. 1–16, 2018.
- [13] A. Krause, J. Leskovec, C. Guestrin, J. VanBriesen, and C. Faloutsos, "Efficient sensor placement optimization for securing large water distribution networks," *J. Water Resour. Planning Manage.*, vol. 134, no. 6, pp. 516–526, 2008.
- [14] J. Wägberg, D. Zachariah, T. B. Schön, and P. Stoica, "Prediction performance after learning in Gaussian process regression," in *Proc. 20th Int. Conf. Artif. Intel. Statist.*, Apr. 20–22, 2017, vol. 54, pp. 1264–1272.
- [15] E. A. Yfantis, G. T. Flatman, and J. V. Behar, "Efficiency of kriging estimation for square, triangular, and hexagonal grids," *Math. Geol.*, vol. 19, no. 3, pp. 183–205, 1987.
- [16] V. Suryan and P. Tokekar, "Learning a spatial field with Gaussian process regression in minimum time," in *Proc. 13th Int. Workshop Algorithmic Found. Robot.*, 2018, pp. 301–317.
- [17] F. Bian, D. Kempe, and R. Govindan, "Utility based sensor selection," in *Proc. 5th Int. Conf. Inf. Process. Sensor Netw.*, 2006, pp. 11–18.
- [18] W. J. Welch, "Branch-and-bound search for experimental designs based on D optimality and other criteria," *Technometrics*, vol. 24, no. 1, pp. 41–48, 1982.
- [19] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [20] D. S. Hochbaum and W. Maass, "Approximation schemes for covering and packing problems in image processing and VLSI," *J. ACM*, vol. 32, no. 1, pp. 130–136, 1985.
- [21] H. González-Banos, "A randomized art-gallery algorithm for sensor placement," in *Proc. 17th Annu. Symp. Comput. Geometry*, 2001, pp. 232–240.
- [22] W. F. Caselet and J. V. Zidek, "Optimal monitoring network designs," *Statist. Probability Lett.*, vol. 2, no. 4, pp. 223–227, 1984.
- [23] N. Cressie, "Statistics for spatial data," *Terra Nova*, vol. 4, no. 5, pp. 613–617, 1992.
- [24] M. C. Shewry and H. P. Wynn, "Maximum entropy sampling," *J. Appl. Statist.*, vol. 14, no. 2, pp. 165–170, 1987.
- [25] C.-W. Ko, J. Lee, and M. Queyranne, "An exact algorithm for maximum entropy sampling," *Oper. Res.*, vol. 43, no. 4, pp. 684–691, 1995.
- [26] N. Ramakrishnan, C. Bailey-Kellogg, S. Tadepalli, and V. N. Pandey, "Gaussian processes for active data mining of spatial aggregates," in *Proc. SIAM Int. Conf. Data Mining*, 2005, pp. 427–438.
- [27] D. L. Zimmerman, "Optimal network design for spatial prediction, covariance parameter estimation, and empirical prediction," *Environmetrics*, vol. 17, no. 6, pp. 635–652, 2006.
- [28] Z. Zhu and M. L. Stein, "Spatial sampling design for prediction with estimated parameters," *J. Agricultural, Biol. Environ. Statist.*, vol. 11, no. 1, Mar. 2006, Art. no. 24.
- [29] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions—I," *Math. Program.*, vol. 14, no. 1, pp. 265–294, 1978.
- [30] A. C. Atkinson, *Optimal Design*. Amer. Cancer Soc., 2015, pp. 1–17. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118445112.stat04090.pub2>
- [31] L. Van Nguyen, S. Kodagoda, R. Ranasinghe, and G. Dissanayake, "Simulated annealing based approach for near-optimal sensor selection in Gaussian processes," in *Proc. Int. Conf. Control, Autom. Inf. Sci.*, Nov. 2012, pp. 142–147.
- [32] S. T. Jawaid and S. L. Smith, "Informative path planning as a maximum traveling salesman problem with submodular rewards," *Discrete Appl. Math.*, vol. 186, pp. 112–127, 2015.
- [33] A. Krause and C. Guestrin, "Nonmyopic active learning of Gaussian processes: An exploration-exploitation approach," in *Proc. 24th Int. Conf. Mach. Learn.*, 2007, pp. 449–456.
- [34] K. H. Low, J. Chen, J. M. Dolan, S. Chien, and D. R. Thompson, "Decentralized active robotic exploration and mapping for probabilistic field classification in environmental sensing," in *Proc. 11th Int. Conf. Auton. Agents Multiagent Syst.*, 2012, vol. 1, pp. 105–112.
- [35] F. Galland, P. Réfrégier, and O. Germain, "Synthetic aperture radar oil spill segmentation by stochastic complexity minimization," *IEEE Geosci. Remote Sens. Lett.*, vol. 1, no. 4, pp. 295–299, Oct. 2004.
- [36] A. Singh, R. Nowak, and P. Ramanathan, "Active learning for adaptive mobile sensing networks," in *Proc. 5th Int. Conf. Inf. Process. Sensor Netw.*, 2006, pp. 60–68.
- [37] K. Dantu and G. S. Sukhatme, "Detecting and tracking level sets of scalar fields using a robotic sensor network," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2007, pp. 3665–3672.
- [38] S. Srinivasan, K. Ramamritham, and P. Kulkarni, "Ace in the hole: Adaptive contour estimation using collaborating mobile sensors," in *Proc. 7th Int. Conf. Inf. Process. Sensor Netw.*, 2008, pp. 147–158.
- [39] B. Bryran, R. C. Nichol, C. R. Genovese, J. Schneider, C. J. Miller, and L. Wasserman, "Active learning for identifying function threshold boundaries," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 163–170.
- [40] B. Bryran and J. Schneider, "Actively learning level-sets of composite functions," in *Proc. 25th Int. Conf. Mach. Learn.*, 2008, pp. 80–87.
- [41] A. Gotovos, N. Casati, G. Hitz, and A. Krause, "Active learning for level set estimation," in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, 2013, pp. 1344–1350.
- [42] M. Wallat, T. Graepel, and K. Obermayer, "Gaussian process regression: Active data selection and test point rejection," in *Proc. IEEE-INNS-ENNS Int. Joint Conf. Neural Netw., Neural Comput.: New Challenges Perspectives New Millennium*, Jul. 2000, vol. 3, pp. 241–246.
- [43] A. Krause and C. Guestrin, "Submodularity and its applications in optimized information gathering," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 4, 2011, Art. no. 32.
- [44] C. Chekuri and M. Pal, "A recursive greedy algorithm for walks in directed graphs," in *Proc. 46th Annu. IEEE Symp. Found. Comput. Sci.*, 2005, pp. 245–253.
- [45] J. Binney, A. Krause, and G. S. Sukhatme, "Optimizing waypoints for monitoring spatiotemporal phenomena," *Int. J. Robot. Res.*, vol. 32, no. 8, pp. 873–888, 2013.
- [46] A. Das and D. Kempe, "Algorithms for subset selection in linear regression," in *Proc. 40th Annu. ACM Symp. Theory Comput.*, 2008, pp. 45–54.
- [47] B. Zhang and G. S. Sukhatme, "Adaptive sampling for estimating a scalar field using a robotic boat and a sensor network," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2007, pp. 3673–3680.
- [48] N. E. Leonard, D. A. Paley, F. Lekien, R. Sepulchre, D. M. Fratantoni, and R. E. Davis, "Collective motion, sensor networks, and ocean sampling," *Proc. IEEE*, vol. 95, no. 1, pp. 48–74, Jan. 2007.
- [49] D. O. Popa, M. F. Mysorewala, and F. L. Lewis, "EKF-based adaptive sampling with mobile robotic sensor nodes," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2006, pp. 2451–2456.
- [50] R. Sim and N. Roy, "Global A-optimal robot exploration in SLAM," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2005, pp. 661–666.
- [51] M. Rahimi, R. Pon, W. J. Kaiser, G. S. Sukhatme, D. Estrin, and M. Srivastava, "Adaptive sampling for environmental robotics," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2004, vol. 4, pp. 3537–3544.
- [52] K. H. Low, G. J. Gordon, J. M. Dolan, and P. Khosla, "Adaptive sampling for multi-robot wide-area exploration," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2007, pp. 755–760.
- [53] A. Tuchscherer, "Sampling," *Biometrical J.*, vol. 35, no. 7, pp. 848–848, 1993.
- [54] K. H. Low, J. M. Dolan, and P. Khosla, "Adaptive multi-robot wide-area exploration and mapping," in *Proc. 7th Int. Joint Conf. Auton. Agents Multiagent Syst.*, 2008, pp. 23–30.
- [55] K. H. Low, J. M. Dolan, and P. Khosla, "Information-theoretic approach to efficient adaptive path planning for mobile robotic environmental sensing," in *Proc. 19th Int. Conf. Autom. Planning Scheduling*, 2009, pp. 233–240.
- [56] R. Garnett, Y. Krishnamurthy, X. Xiong, J. Schneider, and R. Mann, "Bayesian optimal active search and surveying," in *Proc. 29th Int. Conf. Mach. Learn.*, 2012, pp. 843–850.
- [57] N. K. Yilmaz, C. Evangelinos, P. F. Lermusiaux, and N. M. Patrikalakis, "Path planning of autonomous underwater vehicles for adaptive sampling using mixed integer linear programming," *IEEE J. Ocean. Eng.*, vol. 33, no. 4, pp. 522–537, Oct. 2008.
- [58] N. Cao, K. H. Low, and J. M. Dolan, "Multi-robot informative path planning for active sensing of environmental phenomena: A tale of two algorithms," in *Proc. Int. Conf. Auton. Agents Multi-Agent Syst.*, 2013, pp. 7–14.

- [59] Y. Teck Tan, A. Kunapareddy, and M. Kobilarov, "Gaussian process adaptive sampling using the cross-entropy method for environmental sensing and monitoring," in *Proc. Int. Conf. Robot. Autom.*, 2018, pp. 6220–6227.
- [60] N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger, "Information-theoretic regret bounds for Gaussian process optimization in the bandit setting," *IEEE Trans. Inf. Theory*, vol. 58, no. 5, pp. 3250–3265, May 2012.
- [61] E. Snelson and Z. Ghahramani, "Sparse Gaussian processes using pseudo-inputs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 1257–1264.
- [62] L. Csató and M. Opper, "Sparse on-line Gaussian processes," *Neural Comput.*, vol. 14, no. 3, pp. 641–668, 2002.
- [63] M. Seeger, C. Williams, and N. Lawrence, "Fast forward selection to speed up sparse Gaussian process regression," *Artif. Intell. Statist.*, 2003. [Online]. Available: <http://infoscience.epfl.ch/record/161318>
- [64] T. N. Hoang, Q. M. Hoang, and K. H. Low, "A unifying framework of anytime sparse Gaussian process regression models with stochastic variational inference for big data," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 569–578. [Online]. Available: <http://dl.acm.org/citation.cfm?id=3045118.3045180>
- [65] R. Mishra, M. Chitre, and S. Swarup, "Online informative path planning using sparse Gaussian processes," in *Proc. OCEANS-MTS/IEEE Kobe Techno-Oceans*, 2018, pp. 1–5.
- [66] W. Luo, S. S. Khatib, S. Nagavalli, N. Chakraborty, and K. Sycara, "Distributed knowledge leader selection for multi-robot environmental sampling under bandwidth constraints," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 5751–5757.
- [67] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Trans. Robot. Autom.*, vol. 20, no. 2, pp. 243–255, Apr. 2004.
- [68] W. Luo and K. Sycara, "Adaptive sampling and online learning in multi-robot sensor coverage with mixture of Gaussian processes," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2018, pp. 6359–6364.
- [69] S. Kemna, J. G. Rogers, C. Nieto-Granda, S. Young, and G. S. Sukhatme, "Multi-robot coordination through dynamic Voronoi partitioning for informative adaptive sampling in communication-constrained environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 2124–2130.
- [70] R. Zlot, A. Stentz, M. B. Dias, and S. Thayer, "Multi-robot exploration controlled by a market economy," in *Proc. IEEE Int. Conf. Robot. Autom. (Cat. no. 02CH37292)*, 2002, vol. 3, pp. 3016–3023.
- [71] R. G. Simmons *et al.*, "Coordination for multi-robot exploration and mapping," in *Proc. 17th Nat. Conf. Artif. Intell. 12th Conf. Innovative Appl. Artif. Intell.*, 2000, pp. 852–858.
- [72] W. Sheng, Q. Yang, S. Ci, and N. Xi, "Multi-robot area exploration with limited-range communications," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IEEE Cat. no. 04CH37566)*, 2004, vol. 2, pp. 1414–1419.
- [73] D. E. Soltero, M. Schwager, and D. Rus, "Generating informative paths for persistent sensing in unknown environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 2172–2179.
- [74] A. Marino, G. Antonelli, A. P. Aguiar, A. Pascoal, and S. Chiaverini, "A decentralized strategy for multirobot sampling/patrolling: Theory and experiments," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 1, pp. 313–322, Jan. 2015.
- [75] A. Meliou, A. Krause, C. Guestrin, and J. M. Hellerstein, "Nonmyopic informative path planning in spatio-temporal models," in *Proc. 22nd Nat. Conf. Artif. Intell.*, 2007, pp. 602–607.
- [76] G. A. Hollinger, B. Englot, F. Hover, U. Mitra, and G. S. Sukhatme, "Uncertainty-driven view planning for underwater inspection," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2012, pp. 4884–4891.
- [77] S. T. O'Callaghan and F. T. Ramos, "Gaussian process occupancy maps," *Int. J. Robot. Res.*, vol. 31, no. 1, pp. 42–62, 2012.
- [78] M. C. Kennedy and A. O'Hagan, "Bayesian calibration of computer models," *J. Roy. Statist. Soc.: Series B (Statist. Methodology)*, vol. 63, no. 3, pp. 425–464, 2001.
- [79] J. E. Oakley and A. O'Hagan, "Probabilistic sensitivity analysis of complex models: a Bayesian approach," *J. Roy. Statist. Soc.: Series B (Statist. Methodology)*, vol. 66, no. 3, pp. 751–769, 2004.
- [80] T. Hastie, R. Tibshirani, J. Friedman, and J. Franklin, "The elements of statistical learning: Data mining, inference and prediction," *Math. Intell.*, vol. 27, no. 2, pp. 83–85, 2005.
- [81] M. L. Stein, *Interpolation of Spatial Data: Some Theory for Kriging*. Berlin, Germany: Springer, 2012.
- [82] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Hoboken, NJ, USA: Wiley, 2012.
- [83] O. Tekdas and V. Isler, "Sensor placement for triangulation-based localization," *IEEE Trans. Autom. Sci. Eng.*, vol. 7, no. 3, pp. 681–685, Jul. 2010.
- [84] S. Arora, "Nearly linear time approximation schemes for Euclidean TSP and other geometric problems," in *Proc. 38th Annu. Symp. Found. Comput. Sci.*, Oct. 1997, pp. 554–563.
- [85] O. Tekdas, D. Bhadauria, and V. Isler, "Efficient data collection from wireless nodes under the two-ring communication model," *Int. J. Robot. Res.*, vol. 31, no. 6, pp. 774–784, 2012.
- [86] G. N. Frederickson, M. S. Hecht, and C. E. Kim, "Approximation algorithms for some routing problems," in *Proc. 17th Annu. Symp. Found. Comput. Sci.*, Oct. 1976, pp. 216–227.
- [87] D. J. Mulla, A. C. Sekely, and M. Beatty, "Evaluation of remote sensing and targeted soil sampling for variable rate application of nitrogen," in *Proc. 5th Int. Conf. Precision Agriculture*, 2000, pp. 1–15.
- [88] C. E. Rasmussen and H. Nickisch, "Gaussian processes for machine learning (GPML) toolbox," *J. Mach. Learn. Res.*, vol. 11, pp. 3011–3015, 2010.



Varun Suryan (Student Member, IEEE) received the B.Tech. degree in mechanical engineering from the Indian Institute of Technology Jodhpur, Jodhpur, India, in 2016 and the M.S. degree in computer engineering from Virginia Tech, Blacksburg, VA, USA, in 2019. He is pursuing the Ph.D. degree in computer science with the University of Maryland, College Park, MD, USA.

His research interests include algorithmic robotics, Gaussian processes, and reinforcement learning.



Pratap Tokekar (Member, IEEE) received the Bachelor of Technology degree in electronics and telecommunication from the College of Engineering Pune, Pune, India, in 2008, and the Ph.D. degree in computer science from the University of Minnesota, Minneapolis, MN, USA, in 2014.

Previously, he was a Postdoctoral Researcher with the GRASP Lab of University of Pennsylvania, Philadelphia, PA, USA. Between 2015 and 2019, he was an Assistant Professor with the Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, USA. Currently, he is an Assistant Professor with the Department of Computer Science, University of Maryland, College Park, MD, USA.

Dr. Tokekar was a recipient of the NSF CISE Research Initiation Initiative award. He is an Associate Editor for the *IEEE Robotics and Automation Letters* and *IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING*.