An Improved Greedy Algorithm for Subset Selection in Linear Estimation

Shamak Dutta, Nils Wilde, and Stephen L. Smith

Abstract— In this paper, we consider a subset selection problem in a spatial field where we seek to find a set of k locations whose observations provide the best estimate of the field value at a finite set of prediction locations. The measurements can be taken at any location in the continuous field, and the covariance between the field values at different points is given by the widely used squared exponential covariance function. One approach for observation selection is to perform a grid discretization of the space and obtain an approximate solution using the greedy algorithm. The solution quality improves with a finer grid resolution but at the cost of increased computation. We propose a method to reduce the computational complexity, or conversely to increase solution quality, of the greedy algorithm by considering a search space consisting only of prediction locations and centroids of cliques formed by the prediction locations. We demonstrate the effectiveness of our proposed approach in simulation, both in terms of solution quality and runtime.

I. INTRODUCTION

An important problem in engineering applications is deciding the subset of measurements that are the most useful in the estimation of an unknown quantity of interest. For example, in agriculture, it is important to estimate the nutrient quality of a field using soil samples. This helps guide fertilizer usage to replenish lost nutrients, which subsequently maximizes crop yield. However, it is impractical to sample the soil at each location in large agricultural fields. The goal is to determine where to sample the soil, such that the nutrient quality at a large set of prediction locations can be estimated accurately. An example of the soil pH variability in a field with a set of prediction locations is shown in Figure 1. This type of subset selection problem shows up in domains such as sensor placement/active sampling in spatial statistics [1], [2], [3], [4], [5], feature selection in machine learning [6], [7], informative path planning in robotics [8], [9], [10], [11], among others. The challenge is similar: choose the subset of attributes that best estimates the quantity of interest.

The Bayesian approach is to model the quantities as random variables. The estimates of prediction variables are obtained by linear estimators and the objective is to minimize the mean squared estimation error. A benefit of this approach is that prior statistical knowledge of the quantities can be incorporated into the estimation procedure. In addition, the mean squared error resulting from a linear estimator is independent of the observations. Thus, deciding the subset

The authors are with the Department of Electrical and Computer Engineering, University of Waterloo, Canada {stephen.smith, nwilde, shamak.dutta}@uwaterloo.ca. N. Wilde is also with the Cognitive Robotics Department, Delft University of Technology, Netherlands.
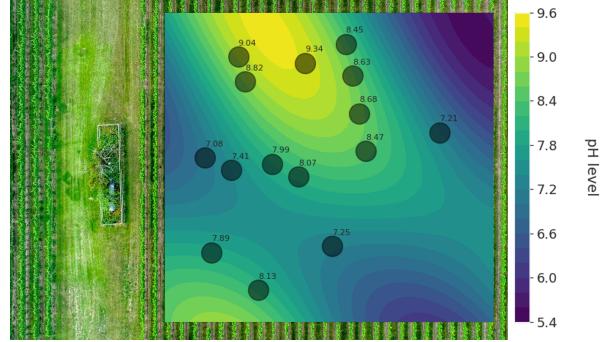
Fig. 1. An example of the pH variability in an agricultural field. The circles are the prediction locations where accurate estimates are desired. Agricultural fields can be large and one can only take a fixed number of soil samples to best estimate the pH variability at the prediction locations.

that minimizes the mean squared error can be done *a priori*. With a finite observation set, a popular approach is to use the greedy algorithm [6], [12]. At each step, the variable maximizing the marginal gain is selected. Continuous observation sets, such as agricultural fields, can be made finite by a grid discretization, which can be used by the greedy algorithm, which we refer to as GRID-GREEDY. The solution quality improves with a finer grid but at an increased computational cost. The objective of this paper is to remove the dependence on the grid discretization while obtaining good solution quality. Our proposed method, CENTROID-GREEDY, restricts the search to the set of prediction locations and the centroids of the cliques formed by the prediction locations. This is motivated by our analysis in one dimension where we identify a critical distance between points that characterizes the optimal measurement location. In our experiments, we show CENTROID-GREEDY achieves better solutions when given the same computational resources as GRID-GREEDY and finds solutions of similar quality more efficiently.

Related Work: Similar to our work, [11] studies the problem of minimizing the number of measurements taken while keeping the estimation error under a threshold in a spatial field modeled as a Gaussian Process with a squared exponential covariance function. In contrast, we minimize the estimation error over a finite set of prediction variables using k observation variables. In addition, the variables are not restricted to be Gaussian in our problem setup.

A seminal paper in sensor placement for Gaussian Processes is [1], which considers maximizing the mutual information between the sensed and unsensed locations. Using the submodularity of mutual information, the authors use a

greedy algorithm to obtain a constant factor approximation guarantee. However, to the best of our knowledge, there is no direct relation between the mutual information and the resulting estimation error. Our work considers the estimation error directly. In [13], the author studies the problem of estimation using kriged Kalman filtering in a spatio-temporal field. In our work, we consider spatial variation only and share the kriging aspect with the work in [13].

Another related problem is the subset selection problem in linear regression, where one has to select a subset of k random variables to yield the best prediction of another random variable of interest. The authors in [14] study a special case where the random variables can be embedded onto the real line with covariances that decay with the distance. They provide an algorithm to compute the optimal solution using dynamic programming. However, their work assumes a finite set of measurement variables, which must be embedded onto the real line. There is no known extension of this idea to an infinite set of observation variables embedded in higher dimensional spaces. In this paper, we consider random variables indexed on an infinite subset of a d-dimensional Euclidean space with the widely used squared exponential covariance function [15]. In [16], the authors introduce the concept of approximate submodularity to provide guarantees for the greedy algorithm. The setup is the same as in [14] with a finite set of observation variables. Thus, the performance guarantees are not directly applicable.

Contributions: The contributions of this work are twofold. First, we formulate a problem of budget constrained observation selection from an infinite set to best estimate a finite set of prediction variables. Second, we propose CENTROID-GREEDY, a greedy algorithm that uses a ground set consisting of the prediction locations and the centroids of cliques formed by the prediction locations. This reduces the dependence of GRID-GREEDY on the grid discretization of the continuous field. In simulations, we demonstrate the improved solution quality and run time of CENTROID-GREEDY in comparison to GRID-GREEDY. Due to space limitations, proofs are omitted and can be found in the preprint [17].

II. PRELIMINARIES

Many combinatorial problems involve maximizing a submodular set function whose definition follows.

Definition 1 (Submodular Set Function). Given a finite set $\mathcal{V}$, the set function $f : 2^{\mathcal{V}} \to \mathbb{R}$ is submodular if for all sets $A \subseteq B \subseteq \mathcal{V}$ and $x \in \mathcal{V} \setminus B$, the diminishing returns property is satisfied:

$$f(A \cup \{x\}) - f(A) \geq f(B \cup \{x\}) - f(B). \quad (1)$$

The greedy algorithm for maximizing set functions subject to a cardinality constraint is described as follows.

Definition 2 (Greedy Algorithm). The greedy algorithm begins with the empty set $S_0 = \emptyset$ and repeatedly adds the element $x \in V$ that maximizes the marginal gain until the cardinality constraint is met. That is, for $i \geq 1$,

$$S_{i+1} = S_i \cup \{\arg\max_{x \in V} f(S_i \cup \{x\}) - f(S_i)\}. \quad (2)$$

The cardinality constrained maximization of a certain class of submodular functions can be efficiently approximated by a greedy algorithm whose solution is within a multiplicative factor of $1 - 1/e \approx 0.63$ of the optimal [18].

Let $X_1, \ldots, X_n, Y$ be square integrable, zero mean random variables, and $\boldsymbol{b} := (\mathrm{Cov}(X_1, Y), \ldots, \mathrm{Cov}(X_n, Y))^T$, $\boldsymbol{X} := (X_1, \ldots, X_n)^T$, and let C be a $n \times n$ matrix whose $(i, j)^{\text{th}}$ element is $\mathrm{Cov}(X_i, X_j)$.

Definition 3 (Linear Least Squares Estimator). Given $X_1, \ldots, X_n$, the optimal linear estimator of Y is:

$$\hat{Y} := \boldsymbol{b}^T C^{-1} \boldsymbol{X}. \quad (3)$$

Definition 4 (Mean Squared Estimation Error). Given $X_1, \ldots, X_n$, the linear least squares estimator of Y results in a mean squared estimation error of

$$\mathbb{E}\left[\left(Y - \hat{Y}\right)^2\right] := \mathrm{Var}(Y) - \boldsymbol{b}^T C^{-1} \boldsymbol{b}. \quad (4)$$

III. PROBLEM FORMULATION

Let $D \subset \mathbb{R}^d$ represent the environment and let $\sigma_0 \in \mathbb{R}_{>0}$ be a positive real number. For any location $x \in D$, let $Z(x)$ be a random variable with zero mean and variance σ_0^2. We consider a convex set of measurement locations $\Theta \subset D$ and a finite set of prediction locations $\Omega \subset \Theta$. Given a positive integer $k \in \mathbb{Z}_+$, our goal is to minimize the mean-squared error of the linear estimation of the prediction variables $\{Z(x) : x \in \Omega\}$ using only k measurement variables.

Remark. Note that when $|\Omega| < k$, measurements at all prediction locations will yield low estimation error. The problem is only interesting when $|\Omega| > k$.

We define $\phi_{\mathrm{SE}} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{>0}$ to be the squared exponential covariance function with known parameters σ_0 and $L \in \mathbb{R}_{>0}$:

$$\phi_{\mathrm{SE}}(x) = \sigma_0^2 e^{-\frac{x^2}{2L^2}}, \quad (5)$$

The parameters can be learned from a pilot deployment or expert knowledge and is a standard assumption in sensor placement algorithms [1].

For any $x, y \in D$, we assume the covariance of the random variables $Z(x), Z(y)$ is given by

$$\mathrm{Cov}(Z(x), Z(y)) = \mathbb{E}[Z(x)Z(y)] = \phi_{\mathrm{SE}}(\|x - y\|). \quad (6)$$

Let i be a positive integer. For any $x \in D$, let $Y_i(x)$ be the ith noisy measurement of $Z(x)$ and let the associated noise be $\epsilon_i(x)$. The noise is assumed to be a zero mean random variable with variance $\sigma^2 > 0$. In addition, the noise is uncorrelated across measurements and locations i.e. for any $x, y \in D$ and for any positive integers $m, n \in \mathbb{Z}_+$, $\mathrm{Cov}(\epsilon_m(x), \epsilon_n(y)) = 0$. The measurement is

$$Y_i(x) = Z(x) + \epsilon_i(x). \quad (7)$$

In order to reduce notational clutter, for any $x \in D$, we drop the subscript i from the measurement variable $Y_i(x)$ and the associated noise $\epsilon_i(x)$. If there are multiple measurements at the same location, the associated noise terms are uncorrelated. In addition, any measurement at the

same location $x \in D$ (even if there are multiple) will be denoted by $Y(x)$. Now, the measurement equation is

$$Y(x) = Z(x) + \epsilon(x). \qquad (8)$$

We wish to minimize the total mean-squared error, which gives us the following constrained optimization problem:

$$\min_{S \subset \Theta, |S| \leq k} \sum_{x \in \Omega} \mathbb{E}\left[\left(Z(x) - \hat{Z}(x,S)\right)^2\right], \qquad (9)$$

where $\hat{Z}(x,S)$ is the linear estimator of $Z(x)$ given the variables in S. We now rewrite the problem as follows. Denote the elements of a set S by $\{x_1, \ldots, x_k\}$. The linear estimator $\hat{Z}(x,S)$ is given by Definition 3:

$$\hat{Z}(x,S) := \boldsymbol{b}_x(S)^T C(S)^{-1} \boldsymbol{Y}_S \qquad (10)$$

where

$$\boldsymbol{b}_x(S) := [\phi_{\text{SE}}(\|x - x_1\|), \ldots, \phi_{\text{SE}}(\|x - x_k\|)] \in \mathbb{R}^k$$
$$\boldsymbol{Y}_S := [Y(x_1), \ldots, Y(x_k)] \in \mathbb{R}^k$$
$$C(S) := \mathbb{E}\left[\boldsymbol{Z}_S \boldsymbol{Z}_S^T\right] + \sigma^2 I_k \in \mathbb{R}^{k \times k}$$
$$= \begin{bmatrix} \phi_{\text{SE}}(0) & \cdots & \phi_{\text{SE}}(\|x_1 - x_k\|) \\ \vdots & \ddots & \vdots \\ \phi_{\text{SE}}(\|x_k - x_1\|) & \cdots & \phi_{\text{SE}}(0) \end{bmatrix} \qquad (11)$$
$$+ \sigma^2 I_k.$$

Using Definition 4, we can rewrite Equation (9) as

$$\min_{S \subset \Theta, |S| \leq k} \sum_{x \in \Omega} \phi_{\text{SE}}(0) - \boldsymbol{b}_x(S)^T C(S)^{-1} \boldsymbol{b}_x(S). \qquad (12)$$

Since $\phi_{\text{SE}}(0) = \sigma_0^2$ is a constant, we can consider the maximization version of the problem. Define

$$f_x(S) := \boldsymbol{b}_x(S)^T C(S)^{-1} \boldsymbol{b}_x(S)$$
$$f(S) := \sum_{x \in \Omega} f_x(S), \qquad (13)$$

where $\boldsymbol{b}_x(S)$ and $C(S)$ are defined in (11). The function $f_x(S)$ is also known as the *squared multiple correlation* [6], [14] or the *variance reduction* [19].

In this paper, we seek to find the measurement set that maximizes the total variance reduction.

Problem 1. Given measurement locations Θ, prediction locations Ω, and a budget $k > 0$, find a measurement set $S \subset \Theta$ of size k that maximizes the total variance reduction:

$$\max_{S \subset \Theta, |S| \leq k} f(S). \qquad (14)$$

IV. PROBLEM STRUCTURE

In this section we provide preliminary results that guide the design of our algorithm, presented in the next section.

A. Non-submodularity

Problem 1 resembles a sensor placement problem where one is interested in a subset of locations to deploy sensors to maximize the information gained about the environment. Metrics related to the information gained such as coverage and mutual information are known to be submodular functions which can be approximately solved efficiently with a guarantee. However, for Problem 1, we provide an example to show the variance reduction objective is *not submodular*.

Example 1. Consider the following environment setup where the points lie on an interval on the real line.

$$D \subset \mathbb{R}, \Omega = \{0\}, \Theta = [0,2], \sigma = 1, \sigma_0 = 1, L = 1,$$
$$A = \{0.6784\}, B = \{0.6784, 1.4869\}, x = 0.6892. \qquad (15)$$

Now, $f(A \cup \{x\}) - f(A) = 0.1021$ and $f(B \cup \{x\}) - f(B) = 0.1025$, which shows the violation. $\qquad \square$

B. Two Prediction Locations with One Sample

In this subsection, we discuss properties of the problem in 1-D, i.e., the random variables are associated with locations on the real line. This restriction provides valuable insight into the problem and motivates our proposed algorithm.

Suppose the set of prediction locations contains two points i.e. $\Omega = \{y_1, y_2\} \subset [a, b]$, with $y_1 < y_2$. After some simplification, the optimization problem in (14) is

$$\max_{x \in [a,b]} \frac{1}{\sigma_0^2 + \sigma^2} \left(\phi_{\text{SE}}^2(\|x - y_1\|) + \phi_{\text{SE}}^2(\|x - y_2\|)\right)$$
$$= \max_{x \in [a,b]} \frac{\sigma_0^4}{\sigma_0^2 + \sigma^2} \left(e^{-\frac{1}{L^2}\|x - y_1\|^2} + e^{-\frac{1}{L^2}\|x - y_2\|^2}\right). \qquad (16)$$

The solution depends on the relationship between the distance between the two prediction locations and L, the parameter of the squared exponential covariance function defined in (5). This is formalized in the following proposition. The proof is omitted and can be found in [17].

Proposition 1. Let $D \subset \mathbb{R}$, $\Omega = \{y_1, y_2\} \subset D, \Theta = D$, $t = 1$, and the midpoint $x^* = \frac{y_1 + y_2}{2}$. Denote the optimal solution to (16) by OPT. Then,

$$\text{OPT} = x^* \iff \|y_2 - y_1\| \leq \sqrt{2}L. \qquad (17)$$

When the points are separated by a distance greater than $\sqrt{2}L$, Proposition 1 guarantees the suboptimality of the midpoint. In this case, the prediction locations are reasonable solutions whose performance guarantee is given by the following proposition (see [17] for the proof).

Proposition 2. Given $D \subset \mathbb{R}$, $\Omega = \{y_1, y_2\} \subset D, \Theta = D$, and $k = 1$, when $\|y_2 - y_1\| > \sqrt{2}L$, the point $x = y_1$ is an approximate maximizer to (16) with a guarantee

$$\frac{f(\{y_1\})}{f(\{x^*\})} \geq 0.62, \qquad (18)$$

where x^* is the optimal measurement location.

Propositions 1 and 2 motivate our algorithm design. For two prediction points and one sample in 1D, either the midpoint is optimal or either prediction point is an approximate

solution. This suggests the following idea: restrict the search of the greedy algorithm to the prediction locations and the centroids of the cliques formed by the prediction locations.

V. Algorithm

In this section, we discuss GRID-GREEDY and its limitations, our proposed algorithm CENTROID-GREEDY based on computing centroids of maximal cliques, and provide a reformulation of computing the marginal gains that speeds up the implementation of both greedy algorithms in practice.

A. GRID-GREEDY

The greedy algorithm is popular for subset selection in regression where it is also known as Forward Selection [14], [6], [12]. In this section, we discuss how the greedy algorithm can be used for infinite observation sets. For Problem 1, starting with $S_0 = \emptyset$, the first step of the algorithm computes the maximizer to

$$
\begin{aligned}
S_1 &= \arg\max_{x \in \Theta} f(\{x\}) \\
&= \arg\max_{x \in \Theta} \frac{1}{\sigma^2 + \sigma_0^2} \sum_{y \in \Omega} \phi_{\text{SE}}^2(\|x - y\|) \\
&= \arg\max_{x \in \Theta} \frac{\sigma_0^4}{\sigma^2 + \sigma_0^2} \sum_{y \in \Omega} e^{-\frac{1}{L^2}\|x-y\|^2}.
\end{aligned}
\tag{19}
$$

This function is non-concave and in general, it is difficult to find the global maximum. To tackle this non-concave maximization problem, the set of measurement locations Θ can be uniformly discretized to form a finite set of points $\bar{\Theta} \subset \Theta$. The point $x \in \bar{\Theta}$ with the maximum function value is returned as an approximate solution. This is known as the Uniform Grid method [20]. Each step of the greedy algorithm can be approximately solved using this method. The grid discretization is determined by a positive integer parameter $\rho \geq 1$ which tiles each dimension with ρ points to form $\bar{\Theta}$ of size ρ^d. We refer to this method as GRID-GREEDY. The time complexity of GRID-GREEDY is given in the following proposition (see [17] for the proof).

Proposition 3. Given a positive integer $\rho \geq 1$ and a grid discretization of size ρ^d, GRID-GREEDY finds a solution to Problem 1 in time $O(\rho^d k^3 \max\{k, |\Omega|\})$.

The dependence of the runtime on ρ^d is concerning. To get good quality solutions using the greedy algorithm, ρ needs to be sufficiently large to achieve a good grid resolution. In this paper, we aim to find good quality solutions using the greedy algorithm in time *independent* of the grid discretization.

B. CENTROID-GREEDY

We now present our algorithm CENTROID-GREEDY which involves two parts. First, we find the centroids of maximal cliques in a graph with nodes as prediction locations. Second, we use the set of centroids and prediction locations as a ground set for the greedy algorithm for maximizing set functions (Definition 2) to solve Problem 1.

Algorithm 1: MAXIMALCLIQUECENTROIDS

Input: Prediction locations Ω
Output: Clique Centroids $\mathcal{X} \subset \Theta$
1 $G = (V, E) \leftarrow$ CONSTRUCTGRAPH(Ω)
2 $\mathcal{C} \leftarrow$ MAXIMALCLIQUES(G)
3 Initialize $\mathcal{X} = \emptyset$
4 **for** *each clique* $\mathcal{M} \in \mathcal{C}$ **do**
5 $\quad \mathcal{X} \leftarrow \mathcal{X} \cup \{\text{CENTROID}(\mathcal{M})\}$
6 **return** $\mathcal{X}$
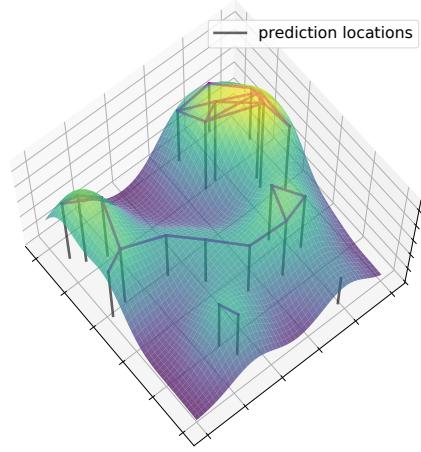
Fig. 2. The objective function $f(S)$ when the budget $k = 1$ for a given set of prediction locations in two dimensions. Two prediction locations are connected by an edge if their distance is less than or equal to $\sqrt{2}L$.

Finding Clique Centroids: The steps to compute clique centroids is given in Algorithm 1. The first step (Line 1, CONSTRUCTGRAPH) constructs a graph $G = (V, E)$ with vertices as prediction locations. Two vertices are connected with an edge if the corresponding prediction locations are within a distance $\sqrt{2}L$. An example of a constructed graph for a two dimensional problem is shown in Figure 2. The next step is to compute the clique centroids. Ideally, we would like to find maximum cliques in the graph. Unfortunately, finding maximum cliques is NP-Hard [21]. We limit ourselves to finding *maximal* cliques from each vertex in the graph since this can be done efficiently with a greedy algorithm: for each vertex $v \in V$ in the graph, grow the clique one vertex at a time by looping through the remaining vertices, add it to the clique if it is adjacent to every vertex in the clique and discard it otherwise (Line 2, MAXIMALCLIQUES). Note, this method does not yield *all* maximal cliques like the Bron-Kerbosch algorithm [22], which has an exponential time complexity in the worst case. Once we have the set of maximal cliques, the final step is to loop through the cliques and compute the centroid of the prediction locations associated with the clique (Line 5).

CENTROID-GREEDY: Proposition 3 ensures that the prediction locations are reasonable approximate solutions when the prediction locations are separated by a distance greater than $\sqrt{2}L$. Instead of GRID-GREEDY which has a runtime

of $O(\rho^d k^3 \max\{k, |\Omega|\})$ for Problem 1 (see Proposition 3), we remove the dependence on ρ^d i.e. the grid discretization, by limiting the search to the set of centroids (computed in Algorithm 1) and the set of prediction locations: $\mathcal{X} \cup \Omega$. The set of centroids is a feasible set for Problem 1 since $\Omega \subset \Theta$ and Θ is a convex set i.e. the set of measurement locations Θ contains the centroids of any subset of prediction locations. Since the number of maximal cliques computed by Algorithm 1 is bounded above by the number of prediction locations, the runtime of CENTROID-GREEDY is $O(|\Omega| k^3 \max\{k, |\Omega|\})$. This is an improvement over the runtime $O(\rho^d k^3 \max\{k, |\Omega|\})$ of GRID-GREEDY, as long as $|\mathcal{X} \cup \Omega| < \rho^d$, which we will show in Section VI, is required for GRID-GREEDY to obtain good solutions for large fields. The steps for CENTROID-GREEDY are given in Algorithm 2.

Algorithm 2: CENTROID-GREEDY

Input: Continuous Field: Θ, Prediction Locations: Ω, budget $k > 0$
Output: Measurement Set: $S \subset \Theta$, $|S| = k$
1   $\mathcal{V} = \text{MAXIMALCLIQUECENTROIDS}(\Omega)$
2   Initialize $S_0 = \emptyset$
3   **for** $i = 1$ *to* k **do**
4      $S_i = S_{i-1} \cup \{\arg\max\limits_{x \in \mathcal{V}} f(S_{i-1} \cup \{x\}) - f(S_{i-1})\}$
5   **return** S_k

C. Implementation of the Greedy Algorithm

Each step of the greedy algorithm requires computing the maximizer of the marginal gain $f(S_i \cup \{x\}) - f(S_i)$ over all feasible x. Computing $f(S)$ in the form in (13) is time consuming and is not amenable to vectorization in NumPy [23] directly. Using Proposition 4, the marginal can be rewritten in a form that can be vectorized, and in practice is much faster to compute. For example, computing the solution for 500 prediction points, ground set size of 400, and a budget of 25 takes ≈ 0.5 seconds with vectorization and ≈ 14 seconds with the non-vectorized version.

Proposition 4. The marginal improvement of $f(S)$ when adding an element x to a set A is given by:

$$f(A \cup \{x\}) - f(A) = T_x \sum_{y \in \Omega} \left(R_{x,y}(A) - \phi_{\text{SE}}^2(x - y) \right)^2,$$
(20)

where $T_x = \left(\sigma_0^2 + \sigma^2 - \boldsymbol{b}_x^T(A) C(A)^{-1} \boldsymbol{b}_x(A) \right)^{-1}$ and $R_{x,y}(A) = \boldsymbol{b}_x(A)^T C(A)^{-1} \boldsymbol{b}_y(A)$.

The proof is omitted and can be found in [17].

VI. EVALUATION

In this section, we provide evidence of two advantages of CENTROID-GREEDY over GRID-GREEDY. First, CENTROID-GREEDY obtains higher quality solutions on problem instances where the run time of both algorithms is comparable. Second, on instances where the solution quality
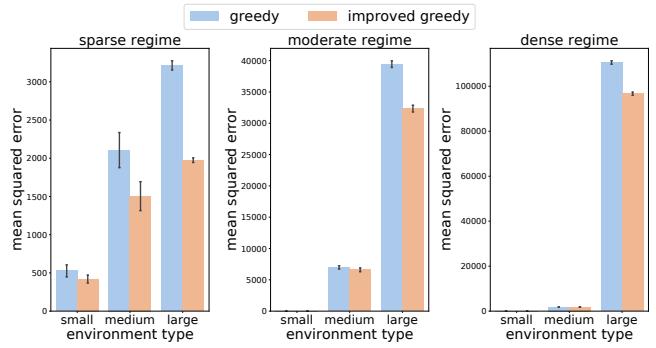
Fig. 3. Comparison of the solution quality when runtimes are kept equal. CENTROID-GREEDY obtains equal or better solutions GRID-GREEDY in all environment types and regime of prediction points.

is comparable, CENTROID-GREEDY finds the solution faster than GRID-GREEDY. The solution quality is measured by the mean squared error (Equation 9) and the run time is measured in seconds.

Experimental Setup: We follow the setup in [11] where a Gaussian Process was fit to a real world dataset of organic matter measurements in an agricultural field [24]. Note that we do not require the actual data, only the parameters of the squared exponential covariance function and the variance of the measurement noise. Specifically, the authors [11] computed $L = 8.33$ meters, $\sigma_0 = 12.87$, and $\sigma^2 = 0.0361$. The interpretation of L is the distance one has to travel before the underlying function value changes [15]. Since the covariance function is a squared exponential, only the relative distances between points matter, not the absolute positions. This enables us to consider different environment sizes:

1) $D_{\text{small}} := \{(x, y) \in \mathbb{R}^2 : 0 \le x \le 40, 0 \le y \le 40\}$, Area = 1600 square meters.
2) $D_{\text{med}} := \{(x, y) \in \mathbb{R}^2 : 0 \le x \le 120, 0 \le y \le 120\}$, Area = 14,400 square meters.
3) $D_{\text{large}} := \{(x, y) \in \mathbb{R}^2 : 0 \le x \le 600, 0 \le y \le 600\}$, Area = 360,000 square meters.

We also consider three regimes for the number of prediction points: sparse (20 points, budget 8), moderate (300 points, budget 75), and dense (1000 points, budget 200). The results in the following sections are based on 10 randomly generated problem instances for each combination of environment type (small, medium, large) and prediction point regime (sparse, moderate, dense). The experiments are implemented using NumPy [23] on an AMD Ryzen 7 2700 processor.

A. Solution Quality

In the first set of experiments, we aim to answer the following question: given the same computational resources, which algorithm provides a better solution? To ensure equal computational resources, for a $N \times N$ grid discretization, we set $N = \left\lceil \sqrt{2|\Omega|} \right\rceil$. Since the number of maximal cliques computed in Algorithm 1 is at most $|\Omega|$, this ensures the runtimes are comparable. The grids selected are: 7×7 (sparse regime), 25×25 (moderate regime), 45×45 (dense regime).

The results are shown in Figure 3. In the sparse regime (left plot) CENTROID-GREEDY outperforms GRID-GREEDY
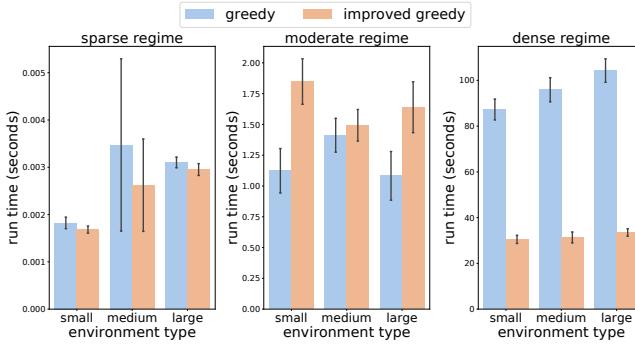
Fig. 4. Comparison of the run time while keeping the solution quality approximately the same. GRID-GREEDY practically takes at least as much time as CENTROID-GREEDY to find solutions of similar quality.

on average in all environment types. The difference in performance is the highest in large environments since the grid resolution is not sufficient to cover the space. In the moderate regime (center plot) and dense regime (right plot), the solution quality of both algorithms is similar in small and medium sized environments. However, for large environments, CENTROID-GREEDY obtains better solutions. The difference in performance reduces as we move from the sparse to dense regime. This is because the high density of prediction points increases the chance of close proximity with grid points, even in the case of low resolution grids.

B. Run Time

In the second set of experiments, we explore how much longer it takes GRID-GREEDY to achieve similar solution quality as CENTROID-GREEDY. For each problem instance, if CENTROID-GREEDY obtains a higher objective value than GRID-GREEDY we repeatedly increase the grid resolution until GRID-GREEDY attains the a similar objective value. We compare the time taken by GRID-GREEDY on the final grid resolution to the time taken by CENTROID-GREEDY.

The results are shown in Figure 4. In the sparse regime (left plot) and moderate regime (center plot), the run times are similar. In the moderate regime, the number of prediction points is a bit higher than the number of grid points and thus the runtime of CENTROID-GREEDY is slightly higher. In the dense regime, GRID-GREEDY takes approximately 2.5 times (small environments), 4 times (medium size environments), and 5 times (large environments) as long as CENTROID-GREEDY to attain a similar objective value. The runtime of GRID-GREEDY increases with the size of the environment, while the runtime of CENTROID-GREEDY remains fairly constant. Note, agricultural fields can be an order of magnitude larger than those considered in this experiment [25] and we expect larger reductions in run time for these agricultural fields in practice.

VII. CONCLUSIONS

We discussed the problem of selecting a k-subset that yields the best linear estimate at a set of prediction locations in a continuous spatial field. We proposed an approach that restricting the search of the greedy algorithm to the set of prediction locations and the centroids of their cliques. This was motivated by identifying a critical distance between two prediction points which characterized the optimal solution in 1D. In simulations, we showed the effectiveness of the proposed approach in terms of solution quality and runtime.

REFERENCES

[1] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies." *Journal of Machine Learning Research*, vol. 9, no. 2, 2008.

[2] N. Ramakrishnan *et al.*, "Gaussian processes for active data mining of spatial aggregates," in *SIAM International Conference on Data Mining*, 2005, pp. 427–438.

[3] J. Le Ny and G. J. Pappas, "On trajectory optimization for active sensing in gaussian process models," in *IEEE Conference on Decision and Control (CDC)*, 2009, pp. 6286–6292.

[4] R. Marchant and F. Ramos, "Bayesian optimisation for intelligent environmental monitoring," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 2242–2249.

[5] S. Yang, N. Wei, S. Jeon, R. Bencatel, and A. Girard, "Real-time optimal path planning and wind estimation using gaussian process regression for precision airdrop," in *American control conference (ACC)*, 2017, pp. 2582–2587.

[6] A. Miller, *Subset selection in regression*. CRC Press, 2002.

[7] I. Guyon *et al.*, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[8] J. Binney and G. S. Sukhatme, "Branch and bound for informative path planning," in *IEEE international Conference on Robotics and Automation*, 2012, pp. 2147–2154.

[9] J. Binney, A. Krause, and G. S. Sukhatme, "Informative path planning for an autonomous underwater vehicle," in *IEEE International Conference on Robotics and Automation*, 2010, pp. 4791–4796.

[10] ——, "Optimizing waypoints for monitoring spatiotemporal phenomena," *The International Journal of Robotics Research*, vol. 32, no. 8, pp. 873–888, 2013.

[11] V. Suryan and P. Tokekar, "Learning a spatial field in minimum time with a team of robots," *IEEE Transactions on Robotics*, vol. 36, no. 5, pp. 1562–1576, 2020.

[12] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, ser. Springer Series in Statistics. New York, NY, USA: Springer New York Inc., 2001.

[13] J. Cortés, "Distributed kriged kalman filter for spatial estimation," *IEEE Trans. on Automatic Control*, vol. 54, pp. 2816–2827, 2009.

[14] A. Das and D. Kempe, "Algorithms for subset selection in linear regression," in *Proceedings of the fortieth annual ACM symposium on Theory of computing*, 2008, pp. 45–54.

[15] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer school on machine learning*. Springer, 2003, pp. 63–71.

[16] A. Das and D. Kempe, "Approximate submodularity and its applications: Subset selection, sparse approximation and dictionary selection," *The Journal of Machine Learning Research*, vol. 19, pp. 74–107, 2018.

[17] S. Dutta, N. Wilde, and S. L. Smith, "An improved greedy algorithm for subset selection in linear estimation," 2022, arXiv:2203.16070 [math.OC].

[18] G. L. Nemhauser *et al.*, "An analysis of approximations for maximizing submodular set functions—i," *Mathematical programming*, vol. 14, no. 1, pp. 265–294, 1978.

[19] A. Krause, H. B. McMahan, C. Guestrin, and A. Gupta, "Robust submodular observation selection." *Journal of Machine Learning Research*, vol. 9, no. 12, 2008.

[20] Y. Nesterov, *Lectures on Convex Optimization*. Springer, 2018.

[21] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. MIT press, 2009.

[22] C. Bron and J. Kerbosch, "Algorithm 457: finding all cliques of an undirected graph," *Communications of the ACM*, vol. 16, no. 9, pp. 575–577, 1973.

[23] C. R. Harris *et al.*, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sept. 2020.

[24] D. Mulla, A. Sekely, and M. Beatty, "Evaluation of remote sensing and targeted soil sampling for variable rate application of nitrogen." in *International Conference on Precision Agriculture*, Bloomington, MN, USA, July 2000, pp. 1–15.

[25] USDA, National Agricultural Statistics Service, "Farms and land in farms, 2019 summary," February 2020. [Online]. Available: https://www.nass.usda.gov/Publications/Todays_Reports/reports/fnlo0220.pdf