A Randomized Greedy Algorithm for Near-Optimal Sensor Scheduling in Large-Scale Sensor Networks

Abolfazl Hashemi, Mahsa Ghasemi, Haris Vikalo, Ufuk Topcu*

Abstract—We study the problem of scheduling sensors in a resource-constrained linear dynamical system, where the objective is to select a small subset of sensors from a large network to perform the state estimation task. We formulate this problem as the maximization of a monotone set function under a matroid constraint. We propose a randomized greedy algorithm that is significantly faster than state-of-the-art methods. By introducing the notion of curvature which quantifies how close a function is to being submodular, we analyze the performance of the proposed algorithm and find a bound on the expected performance of the algorithm in comparison with greedy and semidefinite programming relaxation methods.

I. INTRODUCTION

Large-scale sensor networks have drawn considerable attention in recent years [1]–[6]. In such a network, due to various practical considerations and limitations on resources including computational and communication constraints, the data center which aggregates information typically queries only a small subset of the available sensors. This scenario describes many problems in control systems, signal processing, and machine learning including sensor selection for Kalman filtering [2], [3], [7], batch state estimation and stochastic process estimation [4], [5], minimal actuator placement [8], [9], subset selection in machine learning [10], [11], voltage control and meter placement in power networks [12]–[14], and sensor scheduling in wireless sensor networks [7], [15].

Optimal sensor scheduling requires finding solution to a computationally challenging combinatorial optimization problem, which has motivated development of heuristics and approximate algorithms. In [1], sensor selection is formulated as the maximization (minimization) of the log det of the Fisher information matrix (error covariance matrix) and a semidefinite programming relaxation is proposed. The computational complexity of the SDP relaxation is cubic in the number of sensors in the network which limits practical feasibility of this scheme. Additionally, the SDP relaxation does not come with any performance guarantees. To overcome these drawbacks, Shamaiah et al. [2] proposed a greedy algorithm for the log det maximization formulation of the sensor selection problem whose complexity is significantly lower than that of the SDP relaxation. Since the log det of the Fisher information matrix is a monotone submodular function, the greedy scheme in [2] is a $(1 - 1/e)$-approximation algorithm. More recently, the greedy algorithm for log det maximization was employed and analyzed in a number of practical settings [3]–[5], [8], [9]. For instance, Tzoumas et al. [3] extended the result of [2] by showing that log det of the error covariance matrix of the Kalman filter in presence of process noise is supermodular. However, even though log det is related to the volume of the $\gamma$-confidence ellipsoid, it is not explicitly related to the mean-square error (MSE) which is often the performance measure of interest in sensor scheduling and state estimation problems. The MSE, i.e., the trace of the error covariance matrix, is not supermodular [8], [16], [17]. Therefore, the search for an approximation algorithm with performance guarantees on the estimator’s achievable MSE remains an open research problem.

A. Greedy Algorithms for Combinatorial Optimization

The success of submodular maximization in sensor scheduling applications is due to the extensive prior work in designing approximation algorithms for combinatorial optimization problems. Nemhauser et al. [18] showed that the greedy algorithm provides a $(1 - 1/e)$-approximation to maximizing a monotone submodular set function under cardinality constraint. They extended their work to the nonuniform matroid constraint and showed that the greedy algorithm achieves a $1/(P + 1)$-approximation over the intersection of $P$ matroids [19]. A $(\frac{1}{c} - \frac{1}{c+1})$-approximation algorithm for the monotone submodular function maximization under any matroid constraint was proposed in [20]. A local search method proposed in [21] provides a $(\frac{1}{c} - \frac{1}{c+1})$-approximation for monotone submodular maximization over matroid constraints. Sviridenko et al. [22] introduced a $(1 - c/e - \epsilon)$-approximation algorithm for the maximization of monotone increasing submodular functions and a $(1 + \frac{2}{e} + e^{-1} + \frac{1}{1-e}\Omega(\epsilon))$-approximation algorithm for the minimization of monotone decreasing supermodular functions subject to a single matroid constraint. The approximation ratio there improved the previous $(\frac{1}{c} - \frac{1}{c+1})$-approximation of [20]. In [11], a $(1 - 1/e - \epsilon)$-approximation algorithm is developed for the maximization of monotone increasing submodular functions. Recently, Wang et al. [23] analyzed the performance of the greedy algorithm in the general setting where the function is monotone non-decreasing, but not necessarily submodular. They defined an elemental curvature $\alpha$ and showed that the greedy algorithm provides a $(\frac{1}{1+\mu})$-approximation under a...
general matroid constraint, where \( \mu = \alpha \) if \( 0 \leq \alpha \leq 1 \) and 
\[ \mu = \frac{\alpha^K}{K^1(1-\alpha^K)} \text{ if } \alpha > 1 \] 
\((K' \text{ denotes the rank of the matroid)}\). However, determining the elemental curvature is itself an NP-hard task. Therefore, finding an explicit approximation factor for the settings where the objective function is not supermodular, e.g., trace of the error covariance matrix in sensor scheduling for state estimation via Kalman filtering, remains a challenge.

**B. Contribution**

Processing massive amounts of data collected by large-scale networks may be challenging even when relying on greedy algorithms. Moreover, the objective function that is typically of interest in sensor scheduling applications, the MSE, is not submodular (or supermodular, in case one considers the minimization formulation of the problem). Hence, the performance guarantees for the greedy scheme derived in [18], [19] no longer hold. In this paper, we address these challenges by making the following contributions:

- We formulate the sensor scheduling task as the problem of maximizing a monotone objective function directly related to the MSE of the estimator.
- We propose a randomized greedy algorithm and, by introducing the notion of curvature \( c \), find a bound on the expected MSE of the state estimate formed by the Kalman filter that uses the measurements of the sensors selected by the randomized greedy algorithm.
- We derive a probabilistic bound on the curvature \( c \) for the scenario where the measurements are i.i.d. random vectors with bounded \( \ell_2 \) norm. These results are relevant in the important scenarios of Gaussian and Bernoulli measurement vectors.
- In simulation studies, we illustrate that our proposed randomized greedy algorithm significantly outperforms greedy and SDP relaxation methods in terms of runtime while providing essentially the same or improved MSE.

The rest of the paper is organized as follows. Section II explains the system model and reviews some related concepts. In Section III, we introduce the randomized greedy algorithm and analyze its performance. Section IV presents the simulation results while the concluding remarks are stated in Section V.

**II. SYSTEM MODEL AND PROBLEM FORMULATION**

We briefly summarize the notation used in the paper. Bold capital letters refer to matrices and bold lowercase letters represent vectors. \( A_{ij} \) denotes the \((i,j)\) entry of \( A \). \( a_j \) is the \( j^{th} \) row of \( A \). \( A_S \) is a submatrix of \( A \) that contains rows indexed by set \( S \), and \( \lambda_{\text{max}}(A) \) and \( \lambda_{\text{min}}(A) \) are maximum and minimum eigenvalues of \( A \), respectively. Spectral \((\ell_2)\) norm of a matrix is denoted by \( \| | \|. \) \( I_n \in \mathbb{R}^{n \times n} \) is the identity matrix. Moreover, let \( [n] := \{1, 2, \ldots, n\} \).

**A. Combinatorial Optimization with Matroid Constraint**

This subsection overviews definitions that are essential in the analysis of the randomized greedy algorithm and then introduces the general combinatorial optimization problem studied in the paper.

**Definition II.1.** A set function \( f : 2^X \to \mathbb{R} \) is monotone non-decreasing if \( f(S) \leq f(T) \) for all \( S \subseteq T \subseteq X \).

**Definition II.2.** A set function \( f : 2^X \to \mathbb{R} \) is submodular if
\[
\forall S, T \subseteq X: \quad f(S \cup \{j\}) - f(S) \geq f(T \cup \{j\}) - f(T)
\]
(1)
for all subsets \( S \subseteq T \subseteq X \) and \( j \in X \setminus T \). The term \( f_j(S) = f(S \cup \{j\}) - f(S) \) is the marginal value of adding element \( j \) to set \( S \).

A closely related concept to submodularity is the notion of curvature of a set function that quantifies how close the function is to being submodular. Here, we define the element-wise curvature.

**Definition II.3.** The element-wise curvature of a monotone non-decreasing function \( f \) is defined as
\[
C_l = \max_{(S, T) \in \mathcal{X}_l} \frac{f_i(T)}{f_i(S)}, \quad (2)
\]
where \( \mathcal{X}_l = \{ (S, T, i) | S \subseteq T \subseteq X, i \in X \setminus T, |T| - |S| = l, |X| = n \} \). Furthermore, the maximum element-wise curvature is given by
\[
C_{\text{max}} = \max_{l=1}^{n-1} C_l.
\]

Note that a set function is submodular if and only if \( C_{\text{max}} \leq 1 \).

**Definition II.4.** Let \( X \) be a finite set and let \( \mathcal{I} \) be a collection of subsets of \( X \). The pair \( \mathcal{M} = (X, \mathcal{I}) \) is a matroid if the following properties hold:

- **Hereditary property.** If \( T \in \mathcal{I} \), then \( S \in \mathcal{I} \) for all \( S \subseteq T \).
- **Augmentation property.** If \( S, T \in \mathcal{I} \) and \( |S| < |T| \), then there exists \( e \in T \setminus S \) such that \( S \cup \{e\} \in \mathcal{I} \).

The collection \( \mathcal{I} \) is called the set of independent sets of the matroid \( \mathcal{M} \). A maximal independent set is a basis. It is easy to show that all the bases of a matroid have the same cardinality.

Given a monotone non-decreasing set function \( f : 2^X \to \mathbb{R} \) with \( f(\emptyset) = 0 \), and a uniform matroid \( \mathcal{M} = (X, \mathcal{I}) \), we are interested in the combinatorial problem
\[
\max_{S \in \mathcal{I}} f(S). \quad (3)
\]

Next, we state the problem of sensor scheduling for Kalman filtering and show how to formulate it as a maximization problem described by (3). Then, we present a randomized greedy algorithm to approximately solve this optimization problem and provide its performance guarantees.
B. Sensor Scheduling for Kalman Filtering

Consider a linear time-varying dynamical system and its measurement model,
\[
x(t + 1) = H(t)x(t) + w(t) \\
y(t) = A(t)x(t) + v(t),
\] (4)
where \(x(t) \in \mathbb{R}^m\) is the state vector, \(y(t) \in \mathbb{R}^n\) is the measurement vector, \(w(t)\) and \(v(t)\) are zero-mean Gaussian noises with covariances \(Q(t)\) and \(R(t)\), respectively. \(H(t) \in \mathbb{R}^{m \times m}\) is the state transition matrix and \(A(t) \in \mathbb{R}^{n \times m}\) is the matrix whose rows at time \(t\) are the measurement vectors \(a_i(t) \in \mathbb{R}^n\). We assume that the states \(x(t)\) are uncorrelated with \(w(t)\) and \(v(t)\). In addition, for simplicity of exposition we assume that \(x(0) \sim \mathcal{N}(0, \Sigma_x)\), \(Q(t) = \sigma^2 I_m\), and \(R(t) = \sigma^2 I_n\).

Due to limited resources, fusion center aims to select \(k\) out of \(n\) sensors and use their measurements to estimate the state vector \(x(t)\) by minimizing the mean squared error (MSE) in the Kalman filtering setting. Note that we assume that the measurement vectors \(a_i(t)\) are available at the fusion center.

Let \(P_{t|t-1}\) and \(P_{t|t}\) be the prediction and filtered error covariance at time instant \(t\), respectively. Then
\[
P_{t|t-1} = H(t)P_{t-1|t-1}H(t)^\top + Q(t) \\
P_{t|t} = \left( P_{t|t-1} + A_S(t)^\top R_S(t)^{-1} A_S(t) \right)^{-1},
\]
where \(S_t\) is the set of selected sensors at time \(t\) and \(P_{0|0} = \Sigma_x\). Since \(R(t) = \sigma^2 I_n\) and the measurements are uncorrelated across sensors, it holds that
\[
P_{t|t} = \left( P_{t|t-1} + \sigma^{-2} A_S(t)^\top A_S(t) \right)^{-1} = F_{S_t}^{-1}
\]
where \(F_{S_t} = P_{t|t-1} + \sigma^{-2} \sum_{i \in S_t} a_i(t)a_i(t)^\top\) is the corresponding Fisher information matrix. The MSE at time \(t\) is expressed by the trace of the filtered error covariance matrix \(P_{t|t}\). That is,
\[
\text{MSE} = \mathbb{E} \left[ \| x(t) - \hat{x}_{t|t} \|_2^2 \right] = \text{Tr} \left( F_{S_t}^{-1} \right)
\] (5)
where \(\hat{x}_{t|t}\) denotes the filtered estimate of the state vector at time \(t\).

III. RANDOMIZED GREEDY ALGORITHM FOR SENSOR SCHEDULING

In this section, we propose a randomized greedy algorithm for sensor scheduling in Kalman filtering setting and analyze its performance. It is straightforward to see that scheduling sensors to minimize the MSE objective (5) at each time step \(t\) is equivalent to solving the maximization
\[
\max_{S} \quad \text{Tr} \left( P_{t|t-1} - F_{S_t}^{-1} \right) \\
\text{s.t.} \quad S \subseteq [n], \quad |S| = k.
\] (6)

By defining \(X = [n]\) and \(\mathcal{I} = \{ S \subseteq X \, | \, |S| = k \}\), it is easy to see that \(\mathcal{M} = (X, \mathcal{I})\) is a matroid. Let \(f(S)\) denote the objective function of (6), i.e., inverse of the estimator’s mean-square error. Proposition III.1 below states that \(f(S)\) is monotone and proposes a recursive scheme to efficiently compute the marginal gain of querying a sensor.

**Proposition III.1.** \(f(S)\) is a monotonically increasing set function, \(f(\emptyset) = 0\), and
\[
f_j(S) = \frac{a_j(t)^\top F_{S_t}^{-2} a_j(t)}{\sigma^2 + a_j(t)^\top F_{S_t}^{-1} a_j(t)}
\] (7)
\[
F_{S\cup\{j\}}^{-1} = F_{S_t}^{-1} - \frac{F_{S_t}^{-1} a_j(t)a_j(t)^\top F_{S_t}^{-1}}{\sigma^2 + a_j(t)^\top F_{S_t}^{-1} a_j(t)}
\] (8)

**Proof.** See Appendix I.

The combinatorial optimization problem (6) is NP-hard as one needs to exhaustively search over all schedules of \(k\) sensors to find the optimal solution. An approximate solution, i.e., a schedule of sensors that results in a sub-optimal MSE, can be found by the following SDP relaxation (see Appendix II for the derivations).
\[
\min_{\mathbf{Z}, \mathbf{Y}} \quad \text{Tr} \left( \mathbf{Y} \right) \\
\text{s.t.} \quad 0 \leq z_i \leq 1, \quad \forall i \in [n] \\
\sum_{i=1}^{n} z_i = k \\
\left[ \begin{array}{cc} \mathbf{Y} & \mathbf{A}^{-1} \mathbf{P}^{-1}_{t|t-1} + \sigma^{-2} \sum_{i=1}^{n} z_i a_i(t)a_i(t)^\top \\ \mathbf{I} & \mathbf{F}_{S_t}^{-1} \end{array} \right] \geq 0.
\] (9)

The complexity of the SDP algorithm scales as \(O(n^3)\) which is infeasible in practice. Furthermore, there is no guarantee on the achievable MSE performance of the SDP relaxation. When the number of sensors in a network and the size of the state vector \(x(t)\) are relatively large, even the greedy algorithm proposed in [2] may be computationally prohibitive. To provide practical feasibility, we propose a computationally efficient randomized greedy algorithm (see Algorithm 1) that finds an approximate solution to (6) with a guarantee on the expected MSE. Algorithm 1 performs the task of sensor scheduling in the following way. At each iteration of the algorithm, a subset \(R\) of size \(s\) is sampled uniformly at random and without replacement from the set of sensors. The marginal gain provided by each of these \(s\) sensors to the objective function is computed using (7), and the one yielding the highest marginal gain is added to the set of selected sensors. Then the efficient recursive formula in (8) is used to update \(F_{S_t}^{-1}\) so it can be used in the next iteration. This procedure is repeated \(k\) times.

**Remark:** The parameter \(\epsilon\) in Algorithm 1, \(\epsilon^{-k} \leq \epsilon < 1\), denotes a predefined constant that is chosen to strike a desired balance between performance and complexity. When \(\epsilon = \epsilon^{-k}\), each iteration includes all of the non-selected sensors in \(R\) and Algorithm 1 coincides with the classical greedy scheme. However, as \(\epsilon\) approaches 1, \(|R|\) and thus the overall computational complexity decreases.

A. Performance Analysis of the Proposed Scheme

In this section, we analyze Algorithm 1 and in Theorem III.3 provide a bound on the performance of the proposed
Algorithm 1 Randomized Greedy Sensor Scheduling

1: Input: $P_{t(t-1)}$, $A_t$, $k$, $c$.
2: Output: Subset $S_t \subseteq [n]$ with $|S_t| = k$.
3: Initialize $S_t^{(0)} = 0$, $F_{S_t^{(0)}} = P_{t(t-1)}$.
4: for $i = 0, \ldots, k - 1$
5:     Choose $R$ by sampling $s = \frac{1}{n} \log(1/\epsilon)$ indices uniformly at random from $[n] \setminus S_t^{(i)}$.
6:     $i_s = \arg\max_{j \in R} \sum_{t} (\sum_{j \in R}) a_s(t) a_j(t)$.
7:     Set $S_t^{(i+1)} = S_t^{(i)} \cup \{i_s\}$.
8:     $F_{S_t^{(i+1)}}^{-1} = F_{S_t^{(i)}}^{-1} - \frac{F_{S_t^{(i)}}^{-1} a_i(t) (a_i(t))^{\top} F_{S_t^{(i)}}^{-1}}{\sigma^2 + a_i(t) (a_i(t))^{\top}}$.
9: end for
10: return $S_t = S_t^{(k)}$.

randomized greedy scheme when applied to finding an approximate solution to the maximization (6).

Before stating the main results, we first provide two lemmas. Lemma III.1 upper-bounds the difference between the values of the corresponding to two sets having different cardinalities while Lemma III.2 provides a lower bound on the expected marginal gain.

**Lemma III.1.** Let $\{C_i\}_{i=1}^{n}$ be the element-wise curvatures of $f(S)$. Let $S \subseteq T$ be any schedules of sensors such that $S \subseteq T \subseteq [n]$ with $|T \setminus S| = r$. Then, it holds that

$$f(T) - f(S) \leq C(r) \sum_{j \in T \setminus S} f_j(S),$$

(10)

where $C(r) = \frac{1}{r} (1 + \sum_{i=1}^{r-1} C_i)$.

**Proof.** See Appendix III.

**Lemma III.2.** Let $S_t^{(i)}$ be the set of selected sensors at the end of the $i$th iteration of Algorithm 1. Then

$$\mathbb{E} [f_{(i+1)}(S_t^{(i)} | S_t^{(i)})] \geq \frac{1 - e^\beta}{k} \sum_{j \in O_t \setminus S_t^{(i)}} f_j(S_t^{(i)}),$$

(11)

where $O_t$ is the set of optimal sensors at time $t$, $i_s$ is the index of the selected sensor at the $i$th iteration, and $\beta \geq 1$ is a constant that depends on $n$, $k$, and $c$.

**Proof.** See Appendix IV.

Theorem III.3 below states that Algorithm 1 provides an approximate solution to the sensor scheduling problem. In particular, if $f(S)$ is characterized by a bounded maximum element-wise curvature, Algorithm 1 returns a subset of sensors yielding an objective that is on average within a multiplicative factor of the objective achieved by the optimal schedule.

**Theorem III.3.** Let $C_{max}$ be the maximum element-wise curvature of $f(S)$, i.e., the objective function of sensor scheduling problem in (6). Let $S_i$ denote the schedule of sensors selected by Algorithm 1 at time $t$, and let $O_t$ be the optimum solution of (6) such that $|O_t| = k$. Then $f(S_i)$ is on expectation a multiplicative factor away from $f(O_t)$. That is

$$\mathbb{E} [f(S_t)] \geq \left(1 - \frac{1 - e^\beta}{k} \right) f(O_t),$$

(12)

where $c = 1$ if $C_{max} \leq 1$ and $c = C_{max}$ if $C_{max} \geq 1$, $e^k \leq \epsilon < 1$, and $\beta \geq 1$. Furthermore, the computational complexity of Algorithm 1 is $O(n m^2 \log(\frac{1}{\epsilon}))$ where $n$ is the total number of sensors and $m$ is the dimension of $x_t$.

**Proof.** Consider $S_t^{(i)}$, the set generated at the end of the $i$th iteration of Algorithm 1. Employing Lemma III.1 with $S = S_t^{(i)}$ and $T = O_t \cup S_t^{(i)}$, and using monotonicity of $f$ yields

$$f(O_t) - f(S_t^{(i)}) \leq \left(1 + \sum_{i=1}^{r-1} C_i \right) \frac{1}{r} \left(1 + \sum_{i=1}^{r-1} C_i \right) \leq \sum_{j \in O_t \setminus S_t^{(i)}} f_j(S_t^{(i)}),$$

(13)

where $|O_t \setminus S_t^{(i)}| = r$. Now, using Lemma III.2 we obtain

$$\mathbb{E} [f_{(i+1)}(S_t^{(i)} | S_t^{(i)})] \geq \frac{1 - e^\beta}{k} \left(1 + \sum_{i=1}^{r-1} C_i \right) \mathbb{E} [f(S_t^{(i+1)}) - f(S_t^{(i)})].$$

(14)

Applying the law of total expectation yields

$$\mathbb{E} [f_{(i+1)}(S_t^{(i)})] \geq \frac{1 - e^\beta}{k} \left(1 + \sum_{i=1}^{r-1} C_i \right) \mathbb{E} [f(S_t^{(i)})].$$

(15)

Using the definition of the maximum element-wise curvature, we obtain

$$\frac{1}{r} \left(1 + \sum_{i=1}^{r-1} C_i \right) \leq \frac{1}{r} (1 + (r - 1) C_{max}) = g(r).$$

(16)

It is easy to verify, e.g., by taking the derivative, that $g(r)$ is decreasing (increasing) with respect to $r$ if $C_{max} < 1$ ($C_{max} > 1$). Let $c = 1$ if $C_{max} \leq 1$ and $c = C_{max}$ if $C_{max} \geq 1$. Then

$$\frac{1}{r} \left(1 + \sum_{i=1}^{r-1} C_i \right) \leq \frac{1}{r} (1 + (r - 1) C_{max}) \leq c.$$ (17)

Hence,

$$\mathbb{E} [f(S_t^{(i+1)}) - f(S_t^{(i)})] \geq \frac{1 - e^\beta}{kc} \left( f(O_t) - \mathbb{E} [f(S_t^{(i)})] \right).$$

By induction and due to the fact that $f(\emptyset) = 0$,

$$\mathbb{E} [f(S_t)] \geq \left(1 - \left(1 - \frac{1 - e^\beta}{kc} \right)^k \right) f(O_t).$$

(19)
Finally, using the fact that \((1 + x)^y \leq e^{xy}\) for \(y > 0\) and the easily verifiable fact that \(e^{ax} \leq 1 + axe^a\) for \(0 < x < 1\),

\[
\mathbb{E}[f(S_t)] \geq \left(1 - e^{-\frac{1}{2c} - \frac{c}{4}}\right) f(O_t) \geq \left(1 - e^{-\frac{1}{2c} - \frac{c}{4}}\right) f(O_t).
\]  

(20)

To take a closer look at computational complexity, note that step 6 costs \(O(\frac{m}{n} n^2 \log(\frac{1}{\epsilon}))\) as one needs to compute \(\frac{m}{n} \log(\frac{1}{\epsilon})\) marginal gains, each with complexity \(O(n^2)\).

Step 8 requires \(O(n^2)\) arithmetic operations. Since there are \(k\) such iterations, running time of Algorithm 1 is \(O(nm^2 \log(\frac{1}{\epsilon}))\). This completes the proof. 

Using the definition of \(f(S)\) we obtain Corollary III.3.1 stating that, at each time step, the achievable mean-square error in (5) obtained by forming an estimate using sensors selected by the randomized greedy algorithm is within a factor of the optimal mean-square error.

**Corollary III.3.1.** Let \(C_{\text{max}}\) be the maximum element-wise curvature of \(f(S)\), the objective function of the sensor scheduling problem (6). Let \(\alpha = (1 - e^{-\frac{1}{2c}})\) where \(c = 1\) if \(C_{\text{max}} \leq 1\) and \(c = C_{\text{max}}\) if \(C_{\text{max}} \geq 1\), and \(e^{-k} \leq \epsilon < 1\). Let \(\text{MSE}_{\text{S}}\) denote the mean-square estimation error obtained by forming an estimate using information provided by the sensors selected by Algorithm 1 at time \(t\), and let \(\text{MSE}_{\text{o}}\) be the optimal mean-square error formed using information collected by optimum solution of (6). Then the expected \(\text{MSE}_{\text{S}}\) is bounded as

\[
\mathbb{E}[\text{MSE}_{\text{S}}] \leq \alpha \text{MSE}_{\text{o}} + (1 - \alpha) \text{Tr}(P_{t-1}).
\]

(21)

**Remark:** Since the proposed scheme is a randomized algorithm, Theorem III.3 and Corollary III.3.1 state that the expected MSE associated with the solution returned by Algorithm 1 is a multiplicative factor \(\alpha\) away from the optimal MSE. Notice that, as we expect, \(\alpha\) is decreasing in both \(c\) and \(\epsilon\). If \(f(S)\) is characterized by a small curvature, then \(f(S)\) is nearly submodular and randomized greedy algorithm delivers a near-optimal scheduling. As we decrease \(\epsilon\), \(\alpha\) increases which in turn results in a better approximation factor.

**Remark:** The computational complexity of the greedy method proposed in [2] is \(O(knm^2)\). Hence, our proposed scheme provides a reduction in complexity by \(\frac{k}{\log(\frac{1}{\epsilon})}\) which may be particularly beneficial in large-scale networks.

Recall that \(f(S)\) is not submodular [17]. However, in the statements of Theorem III.3 and Corollary III.3.1 we assumed that it has a bounded maximum element-wise curvature. In Theorem III.4 below, we state that a probabilistic theoretical upper bound on the maximum element-wise curvature of \(f(S)\) exists in scenarios where at each time step the measurement vectors \(a_j(t)\)’s are i.i.d. random vectors.

**Theorem III.4.** Let \(C_{\text{max}}\) be the maximum element-wise curvature of \(f(S)\), i.e., the objective function of sensor scheduling problem. Let \(a_j(t)\) be independent zero-mean random vectors with covariance matrix \(\sigma_j^2 I_m\) such that for each \(j\), \(||a_j(t)||^2_2 \leq C\). Then, for all \(q > 0\) with probability

\[
p \geq 1 - m \exp\left(\frac{-q^2/2}{(C - \sigma^2)(n\sigma^2 + q/3)}\right)
\]

it holds that

\[
C_{\text{max}} \leq \frac{\lambda_{\text{max}}(P_{t-1})^2(\sigma^2 + \lambda_{\text{max}}(P_{t-1}))}{\phi^2(\sigma^2 + \phi C)}.
\]

(22)

where

\[
\phi \geq \left(\frac{1}{\lambda_{\text{min}}(P_{t-1})} + \frac{n\sigma^2 + q}{\sigma^2}\right)^{-1}.
\]

(23)

**Proof.** See Appendix V.

We now consider widely used examples of measurement vectors to show practicality of the results of Theorem III.4.

1) **Multivariate Gaussian measurement vectors:** Let \(a_j(t) \sim \mathcal{N}(0, \frac{1}{m} I_m)\) for all \(j\). It is easy to show that \(\mathbb{E}[||a_j(t)||^2_2] = 1\) for all \(j\). Furthermore, it can be shown that \(||a_j(t)||^2_2\) is with high probability distributed around its expected value. Therefore, for this case, \(\sigma_j^2 = \frac{1}{m}\) and \(C = 1\).

2) **Centered Bernoulli measurement vectors:** Let each entry of \(a_j(t)\) be set to \(\pm\sqrt{\frac{1}{m}}\) with equal probability. Therefore, \(||a_j(t)||^2_2 = 1 = C\). Additionally, \(\sigma_j^2 = \frac{1}{m}\) since the entries of \(a_j(t)\) are i.i.d. zero-mean random variables with variance \(\frac{1}{m}\).

**IV. SIMULATION RESULTS**

To test the performance of the proposed randomized greedy algorithm, we compare it with the classic greedy algorithm and the SDP relaxation (see Section II) in a variety of settings as detailed below.

We consider the problem of Kalman filtering for state estimation in a linear time-varying system. For simplicity, let us assume that the system is in steady state and \(H = I_m\). The initial state is a zero-mean Gaussian random vector with covariance \(\Sigma_0 = I_m\). We further specify zero-mean Gaussian process and measurement noises with covariance matrices \(Q = 0.05 I_m\) and \(R = 0.05 I_m\), respectively. At each time step, the measurement vectors, i.e., the rows of the measurement matrix \(A(t)\), are drawn according to \(\mathcal{N}(0, \frac{1}{m} I_m)\).

The MSE values and running time of each scheme is averaged over 10 Monte-Carlo simulations. The time horizon for each run is \(T = 10\). The greedy and randomized greedy algorithms are implemented in MATLAB while the SDP relaxation scheme is implemented via CVX [24]. All experiments were run on a laptop with 2.0 GHz Intel Core i7-4510U CPU and 8.00 GB of RAM.

We first consider the system having state dimension \(m = 50\), the number of measurements \(n = 400\), and \(k = 55\), and compare the MSE values of each method over the time horizon of interest. For randomized greedy we set \(\epsilon = 0.001\). Fig. 1 shows that the greedy method consistently yields the lowest MSE while the MSE of the randomized greedy algorithm is slightly higher. The MSE performance achieved
by the SDP relaxation is considerably larger than those of
the greedy and randomized greedy algorithms. The running
time of each method is given in Table I. Both the greedy
algorithm and the randomized greedy algorithm are much
faster than the SDP formulation. The randomized greedy
scheme is nearly two times faster than the greedy method.

Note that, in this example, in each iteration of the sensor

| Randomized Greedy | Greedy      | SDP Relaxation |
|-------------------|-------------|---------------|
| 0.20 s            | 0.38 s      | 249.86 s      |

*Table I: Running time comparison of randomized greedy, greedy, and SDP relaxation sensor selection schemes (m = 50, n = 400, k = 55, ϵ = 0.001).*

Fig. 1: MSE comparison of randomized greedy, greedy, and SDP relaxation sensor selection schemes employed in Kalman filtering.

The selection procedure the randomized scheme only computes
the marginal gain for a sampled subset of size 50. As a
comparison, the greedy approach computes the marginal
gain for all 400 sensors. In summary, the greedy method
yields the lowest MSE but is much slower than the proposed
randomized greedy algorithm.

To study the effect of the number of selected sensors on
performance, we vary k from 55 to 115 with increments
of 10. The MSE values at the last time step (i.e., t =
10) for each algorithm are shown in Fig. 2(a). As the
number of selected sensors increases, the estimation becomes
more accurate, as reflected by the MSE of each algorithm.
Further, the difference between the MSE values consistently
decreases as more sensors are selected. The running times
shown in Fig. 2(b) indicate that the randomized greedy
scheme is nearly twice as fast as the greedy method, while
the SDP method is orders of magnitude slower than both
greedy and randomized greedy algorithms.

Finally, we compare the performance of the randomized
greedy algorithm to that of the greedy algorithm as the
size of the system increases. We run both methods for 20
different sizes of the system. The initial size was set to
m = 20, n = 200, and k = 25 and all three parameters
are scaled by β where β varies from 1 to 20. In addition, to
evaluate the effect of ϵ on the performance and runtime of
the randomized greedy approach, we repeat experiments for ϵ ∈
{0.1, 0.01, 0.001}. Note that the computational complexity
SDP relaxation scheme is prohibitive in this setting. Fig. 3(a)
illustrates the percentage difference of the MSE between the
two methods. In particular, we show

\[
\% \Delta \text{MSE} = \frac{\text{MSE}_{\text{RG}} - \text{MSE}_{G}}{\text{MSE}_{G}} \times 100
\]

where “RG” and “G” refer to the randomized greedy and
greedy algorithms, respectively. It can be seen that this
difference between the MSEs reduces as the system scales
up. The running time is plotted in Fig. 3(b). As the figure
illustrates, the gap between the running times grows with
the size of the system and the randomized greedy algorithm
performs nearly 40 times faster than the greedy method for
the largest network. Fig. 3 shows that using a smaller ϵ results
in a lower MSE while it slightly increases the running time.
These results suggest that, for large systems, the randomized
greedy provides almost the same MSE while being much
faster than the greedy algorithm.

**V. CONCLUSION**

In this paper, we considered the problem of state estima-
tion in large-scale linear time-varying dynamical systems.
We proposed a randomized greedy algorithm for selecting
sensors to query such that their choice minimizes the esti-
mator’s mean-square error at each time step. We established
the performance guarantee for the proposed algorithm and
analyzed its computational complexity. To our knowledge,
the proposed scheme is the first randomized algorithm for
sensor scheduling with an explicit bound on its achievable mean-square error. In addition, we provided a probabilistic theoretical bound on the element-wise curvature of the objective function. Furthermore, in simulations we demonstrated that the proposed algorithm is superior to the classical greedy and SDP relaxation methods in terms of running time while providing the same or better utility.

As a future work, we intend to extend this approach to nonlinear dynamical systems and obtain a theoretical guarantee on the quality of the resulting approximate solution found by randomized greedy algorithm. Moreover, it would be of interest to extend the framework established in this manuscript to related problems such as the minimal actuator placement.

REFERENCES

[1] S. Joshi and S. Boyd, “Sensor selection via convex optimization,” IEEE Transactions on Signal Processing, vol. 57, no. 2, pp. 451–462, 2009.
[2] M. Shamaiah, S. Banerjee, and H. Vikalo, “Greedy sensor selection: Leveraging submodularity,” in Decision and Control (CDC), 2010 49th IEEE Conference on, pp. 2572–2577, IEEE, 2010.
[3] V. Tzoumas, A. Jadbabaie, and G. J. Pappas, “Sensor placement for optimal kalman filtering: Fundamental limits, submodularity, and algorithms,” in American Control Conference (ACC), 2016, pp. 191–196, IEEE, 2016.
[4] V. Tzoumas, N. A. Atanasov, A. Jadbabaie, and G. J. Pappas, “Scheduling nonlinear sensors for stochastic process estimation,” arXiv preprint arXiv:1609.08536, 2016.
[5] V. Tzoumas, A. Jadbabaie, and G. J. Pappas, “Near-optimal sensor scheduling for batch state estimation: Complexity, algorithms, and limits,” in Decision and Control (CDC), 2016 IEEE 55th Conference on, pp. 2695–2702, IEEE, 2016.
[6] L. Carlone and S. Karaman, “Attention and anticipation in fast visual-inertial navigation,” arXiv preprint arXiv:1610.03344, 2016.
[7] A. Norcio, A. Tarable, F. Dabbene, and R. Tempo, “Sensor selection and precoding strategies for wireless sensor networks,” IEEE transactions on signal processing, vol. 63, no. 16, pp. 4411–4421, 2015.
[8] T. H. Summers, F. L. Fortes, and J. Lygeros, “On submodularity and controllability in complex dynamical networks,” IEEE Transactions on Control of Network Systems, vol. 3, no. 1, pp. 91–101, 2016.
[9] V. Tzoumas, M. A. Rahimian, G. J. Pappas, and A. Jadbabaie, “Minimal actuator placement with optimal control constraints,” in American Control Conference (ACC), 2015, pp. 2081–2086, IEEE, 2015.
[10] A. Bhaskara, “Greedy column subset selection: New bounds and distributed algorithms,” 2016.
[11] B. Mirzasoleiman, A. Badanidiyuru, A. Karbasi, J. Vondrak, and A. Krause, “Lazier than lazy greedy;” in AAAI Conference on Artificial Intelligence, AAAI, 2015.
[12] M. G. Damavandi, V. Krishnamurthy, and J. R. Martí, “Robust meter placement for state estimation in active distribution systems,” IEEE Transactions on Smart Grid, vol. 6, no. 4, pp. 1972–1982, 2015.
[13] A. Olshevsky, “On (non) supermodularity of average control energy,” IEEE Transactions on Control of Network Systems, 2017.
[14] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, “An analysis of approximations for maximizing submodular set functions I,” Mathematical Programming, vol. 14, no. 1, pp. 265–294, 1978.
[15] G. L. Fisher, G. L. Nemhauser, and L. A. Wolsey, “An analysis of approximations for maximizing submodular set functions II,” in Polyhedral combinatorics, pp. 73–87, Springer, 1978.
[16] Y. Filmus and J. Ward, “Monotone submodular maximization over a matroid via non-oblivious local search,” in Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1134–1148, Society for Industrial and Applied Mathematics, 2015.
[17] Z. Wang, B. Moran, X. Wang, and Q. Pan, “Approximation for submodular and supermodular optimization with bounded curvature,” in Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, vol. 31, no. 1, pp. 191–196, Society for Industrial and Applied Mathematics, 2015.
[18] M. Grant, S. Boyd, and Y. Ye, “Cvx: Matlab software for disciplined convex programming,” 2008.
[19] A. Krause, “Submodularity and curvature: the optimal algorithm,” RMS Kokyuroku Bessatsu B, vol. 23, pp. 253–266, 2010.
[20] Y. Filmus and J. Ward, “On (non) supermodularity of average control energy,” IEEE Transactions on Control of Network Systems, 2017.
[21] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, “An analysis of approximations for maximizing submodular set functions I,” Mathematical Programming, vol. 14, no. 1, pp. 265–294, 1978.
[22] G. L. Fisher, G. L. Nemhauser, and L. A. Wolsey, “An analysis of approximations for maximizing submodular set functions II,” in Polyhedral combinatorics, pp. 73–87, Springer, 1978.
[23] J. Vondrak, “Submodularity and curvature: the optimal algorithm,” RIMS Kokyuroku Bessatsu B, vol. 23, pp. 253–266, 2010.
[24] Y. Filmus and J. Ward, “Monotone submodular maximization over a matroid via non-oblivious local search,” SIAM Journal on Computing, vol. 43, no. 2, pp. 514–542, 2014.
[25] M. Sviridenko, J. Vondrak, and J. Ward, “Optimal approximation for submodular and supermodular optimization with bounded curvature,” in Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1134–1148, Society for Industrial and Applied Mathematics, 2015.
[26] Z. Wang, B. Moran, X. Wang, and Q. Pan, “Approximation for maximizing monotone non-decreasing set functions with a greedy method,” Journal of Combinatorial Optimization, vol. 31, no. 1, pp. 29–43, 2016.
[27] M. Grant, S. Boyd, and Y. Ye, “Cvx: Matlab software for disciplined convex programming,” 2008.
[28] M. Grant, S. Boyd, and Y. Ye, “Cvx: Matlab software for disciplined convex programming,” 2008.
[29] R. Bellman, Introduction to matrix analysis. SIAM, 1997.
[30] A. Krause, “Submodularity and curvature: the optimal algorithm,” RIMS Kokyuroku Bessatsu B, vol. 23, pp. 253–266, 2010.
[31] Y. Filmus and J. Ward, “Monotone submodular maximization over a matroid via non-oblivious local search,” SIAM Journal on Computing, vol. 43, no. 2, pp. 514–542, 2014.
[32] M. Sviridenko, J. Vondrak, and J. Ward, “Optimal approximation for submodular and supermodular optimization with bounded curvature,” in Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1134–1148, Society for Industrial and Applied Mathematics, 2015.
[33] Z. Wang, B. Moran, X. Wang, and Q. Pan, “Approximation for maximizing monotone non-decreasing set functions with a greedy method,” Journal of Combinatorial Optimization, vol. 31, no. 1, pp. 29–43, 2016.
[34] M. Grant, S. Boyd, and Y. Ye, “Cvx: Matlab software for disciplined convex programming,” 2008.
[35] R. Bellman, Introduction to matrix analysis. SIAM, 1997.
[36] A. Krause, “Submodularity and curvature: the optimal algorithm,” RIMS Kokyuroku Bessatsu B, vol. 23, pp. 253–266, 2010.
[37] Y. Filmus and J. Ward, “Monotone submodular maximization over a matroid via non-oblivious local search,” SIAM Journal on Computing, vol. 43, no. 2, pp. 514–542, 2014.
[38] M. Sviridenko, J. Vondrak, and J. Ward, “Optimal approximation for submodular and supermodular optimization with bounded curvature,” in Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1134–1148, Society for Industrial and Applied Mathematics, 2015.
First, note that
\[
f(\emptyset) = \text{Tr} \left( P_{\emptyset t}^{-1} - F_0^{-1} \right) = \text{Tr} \left( P_{\emptyset t}^{-1} - P_{\emptyset t}^{-1} \right) = 0.
\]

Now, for \( j \in [n] \setminus S \) it holds that
\[
\begin{align*}
f_j(S) &= f(S \cup \{ j \}) - f(S) \\
&= \text{Tr} \left( P_{t}^{-1} - F_{S \cup \{ j \}}^{-1} \right) - \text{Tr} \left( P_{t}^{-1} - F_{S}^{-1} \right) \\
&= \text{Tr} \left( F_{S}^{-1} \right) - \text{Tr} \left( F_{S \cup \{ j \}}^{-1} \right) \\
&= \text{Tr} \left( F_{S}^{-1} \right) - \text{Tr} \left( (F_{S} + \sigma^{-2}a_j(t)a_j(t)\top)^{-1} \right) \\
&\overset{(a)}{=} \text{Tr} \left( \frac{F_{S}^{-1}a_j(t)a_j(t)\top}{\sigma^2 + a_j(t)\top F_{S}^{-1}a_j(t)} \right) \\
&\overset{(b)}{=} \frac{a_j(t)\top F_{S}^{-2}a_j(t)}{\sigma^2 + a_j(t)\top F_{S}^{-1}a_j(t)}
\end{align*}
\]

where \((a)\) is by applying matrix inversion lemma (Sherman–Morrison formula) [25] on \((F_{S} + \sigma^{-2}a_j(t)a_j(t)\top)^{-1}\), and \((b)\) is by properties of trace of a matrix. Finally, since \(F_{S}\) is a symmetric positive definite matrix, \(f_j(S) > 0\) which in turn implies monotonicity.

**APPENDIX II**

**DERIVATION OF SDP RELAXATION OF (9)**

Let \( z_i \in \{0, 1\} \) indicate the membership of the \( i \)-th sensor in the selected subset at time \( t \) and define \( z = [z_1, z_2, \ldots, z_n] \top \). Hence, (6) can be written as
\[
\min_{z} \text{Tr} \left( P_{t}^{-1} + \sigma^{-2} \sum_{i=1}^{n} z_i a_i(t) a_i(t)\top \right)^{-1}
\]
\[
\text{s.t. } z_i \in \{0, 1\}, \quad \forall i \in [n] \quad \sum_{i=1}^{n} z_i = k.
\]

The convex relaxation of the above optimization problem is given by
\[
\begin{align*}
\min_{z} \text{Tr} \left( P_{t}^{-1} + \sigma^{-2} \sum_{i=1}^{n} z_i a_i(t) a_i(t)\top \right)^{-1} \\
\text{s.t. } 0 \leq z_i \leq 1, \quad \forall i \in [n] \quad \sum_{i=1}^{n} z_i = k.
\end{align*}
\]

By introducing the PSD matrix \( Y \), (26) can equivalently be written as
\[
\begin{align*}
\min_{z, Y} \text{Tr}(Y) \\
\text{s.t. } 0 \leq z_i \leq 1, \quad \forall i \in [n] \\
\sum_{i=1}^{n} z_i = k \\
Y - \left( P_{t}^{-1} + \sigma^{-2} \sum_{i=1}^{n} z_i a_i(t) a_i(t)\top \right)^{-1} \geq 0
\end{align*}
\]

Note that the expression on the left hand side of last constraint in (27) can be thought of as the Schur complement [26] of the block PSD matrix expresses in the last constraint of (9). Exploiting this idea results in the SDP relaxation of (9).

The solution to the SDP may take fractional values, in which case some kind of sorting and rounding need to be employed in order to obtain the desired solution. Here, we select the sensors corresponding to the \( k \) \( z_i \)'s with largest values.

**APPENDIX III**

**PROOF OF LEMMA III.1**

Let \( S \subset T \) and \( T \setminus S = \{ j_1, \ldots, j_r \} \). Therefore,
\[
\begin{align*}
f(T) - f(S) &= f(S \cup \{ j_1, \ldots, j_r \}) - f(S) \\
&= f_{j_1}(S) + f_{j_2}(S \cup \{ j_1 \}) + \ldots + f_{j_r}(S \cup \{ j_1, \ldots, j_{r-1} \}).
\end{align*}
\]

Applying definition of element-wise curvature yields
\[
\begin{align*}
f(T) - f(S) &\leq f_{j_1}(S) + C_{1} f_{j_2}(S) + \ldots + C_{r-1} f_{j_r}(S) \\
&= f_{j_1}(S) + \sum_{l=1}^{r-1} C_{l} f_{j_l}(S).
\end{align*}
\]

Note that (29) is invariant to the ordering of elements in \( T \setminus S \). In fact, it is straightforward to see that given ordering \( \{ j_1, \ldots, j_r \} \), one can choose a set \( P = \{ P_1, \ldots, P_r \} \) with \( r \) permutations – e.g., by defining the right circular-shift operator \( P_t(\{ j_1, \ldots, j_r \}) = \{ j_{r-t+1}, j_1, \ldots, j_t \} \) for \( 1 \leq t \leq r \) such that \( P_p(j) \neq P_q(j) \) for \( p \neq q \) and \( \forall j \in T \setminus S \). Hence, (29) holds for \( r \) such permutations. Summing all of these \( r \) inequalities we obtain
\[
r(f(T) - f(S)) \leq \left( 1 + \sum_{l=1}^{r-1} C_{l} \right) \sum_{j \in T \setminus S} f_{j_l}(S).
\]

Rearranging (30) yields the desired result.

**APPENDIX IV**

**PROOF OF LEMMA III.2**

First, we aim to bound the probability of the event that the random set \( R \) contains at least an index from the optimal set of sensor as this is a necessary condition to reach the optimal MSE. Consider \( S_t^{(i)} \), the set of selected sensors at the end
of $i$th iteration of Algorithm 1 and let $\Phi = R \cap (O \backslash S_t^{(i)})$. It holds that
\[
\Pr\{\Phi = \emptyset\} = \prod_{l=0}^{s-1} \left(1 - \frac{|O \backslash S_t^{(i)}|}{|[n] \backslash S_t^{(i)}| - l}\right)
\leq \left(1 - \frac{|O \backslash S_t^{(i)}|}{s} \sum_{l=0}^{s-1} \frac{1}{|[n] \backslash S_t^{(i)}| - l}\right)^s
\leq \left(1 - \frac{|O \backslash S_t^{(i)}|}{s} \sum_{l=0}^{s-1} \frac{1}{n - l}\right)^s
\]  
(31)
where (a) is by the inequality of arithmetic and geometric means, and (b) holds since $|[n] \backslash S_t| \leq n$. Now recall for any integer $p$,
\[
H_p = \sum_{l=1}^{p} \frac{1}{l} = \log p + \gamma + \zeta_p
\]  
(32)
where, $H_p$ is the $p$th harmonic number, $\gamma$ is the Euler–Mascheroni constant, and $\zeta_p = \frac{1}{p} - O\left(\frac{1}{p^2}\right)$ is a monotonically decreasing sequence related to Hurwitz zeta function [27]. Therefore, using the identity (32) we obtain
\[
\Pr\{\Phi = \emptyset\} \leq \left(1 - \frac{|O \backslash S_t^{(i)}|}{s} (H_n - H_{n-s})\right)^s
\leq \left(1 - \frac{|O \backslash S_t^{(i)}|}{s} (\log(n - s) + \zeta_n - \zeta_{n-s})\right)^s
\leq \left(1 - \frac{|O \backslash S_t^{(i)}|}{s} (\log(n - s) - \frac{s}{2n(n - s)})\right)^s
\leq \left(1 - \frac{s}{n} e^{\frac{s}{2n(n - s)}} |O \backslash S_t^{(i)}|\right)^s
\]  
(33)
where (a) follows since $\zeta_n - \zeta_{n-s} = \frac{1}{2n} - \frac{1}{2(n - s)} + O\left(\frac{1}{n^2}\right)$, and (b) is by the fact that $(1 + x)^y \leq e^{x y}$ for any real number $y > 0$. Next, the fact that $\log(1 - x) \leq -x - \frac{x^2}{2}$ for $0 < x < 1$ yields
\[
(1 - \frac{s}{n}) e^{\frac{s}{2n(n - s)}} \leq e^{-\frac{s}{2n}}
\]  
(34)
where $\beta = 1 + \left(\frac{1}{2n} - \frac{1}{2(n - s)}\right) \geq 1$ for sufficiently large $n$. Thus,
\[
\Pr\{\Phi \neq \emptyset\} \geq 1 - e^{-\frac{\beta}{2n}} |O \backslash S_t^{(i)}| \geq \frac{1 - \beta}{k} |O \backslash S_t^{(i)}| \geq \frac{1 - \beta}{k} |O \backslash S_t^{(i)}| \]
(35)
by definition of $s$ and the fact that $1 - e^{-\frac{s}{2n}}$ is a concave function. Finally, according to Lemma 2 in [11],
\[
\mathbb{E}[f_{i+1} \mid S_t^{(i)}] \geq \sum_{j \in O_t \backslash S_t^{(i)}} f_o(S_t^{(i)})
\]  
Combining (35) and (36) yields the stated results.

\[\text{Note that without loss of generality and for simplicity we assume that } s \text{ is an integer.}\]