Distributed Online Submodular Maximization in Resource-Constrained Networks

Andrew Clark, Basel Alomair, Linda Bushnell, and Radha Poovendran

***Abstract*—Maximization of submodular set functions arises in wireless applications such as scheduling, caching, and leader selection. For a centralized entity with oracle access to the submodular function, submodular maximization can be approximated up to a constant factor using polynomial-time algorithms; such an entity, however, may be unavailable in decentralized wireless networks. In this paper, we consider maximization of a time-varying submodular function by distributed, resource-constrained nodes. We present algorithms for unconstrained distributed submodular maximization, as well as monotone submodular maximization subject to cardinality constraints. For the unconstrained submodular maximization problem, our algorithm achieves an expected optimality gap of 1/3. For cardinality-constrained submodular maximization, our algorithm achieves an expected optimality gap of 1/2, while reducing the storage and communication overhead, as well as the computation requirements of the nodes, compared to existing techniques. We evaluate our approach through an experimental study using sensor scheduling data, and find that our approach is within ten percent of the best achievable utility in the unconstrained case and within five percent in the constrained case.**

I. Introduction

A variety of network design problems can be formulated as selecting a subset of network nodes to perform a shared task in order to optimize a global objective function. Examples include multi-carrier wireless scheduling [1], wireless content delivery through distributed caching [2], sleep-wake scheduling of sensors [3], data forwarding [4], and leader selection [5]. Since selecting an optimal subset of nodes is a combinatorial problem, and hence is intractable in general, additional properties of the objective function must be identified for scalable network design. One such property that is inherent in many applications [1], [2], [3], [4], [5] is submodularity, a diminishing returns property analogous to concavity of continuous functions. Submodularity leads to polynomial-time algorithms for approximating combinatorial problems up to a provable $O(1)$ optimality gap [6].

Existing algorithms for submodular function maximization [7], [8] typically assume the existence of a centralized entity that has global knowledge of the network and can compute the value of the submodular function at any arbitrary set. Wireless networks, however, often operate in the absence of any centralized authority, leading to the following constraints that are not satisfied by the existing algorithms. First, due to bandwidth and connectivity constraints, wireless nodes have limited network information, and hence cannot evaluate the global objective function for any arbitrary set. Second, changes in network topology due to mobility, as well as environmental factors, lead to time-varying objective functions and hence require online algorithms. Finally, the computation and storage constraints of each node limit the set of feasible algorithms.

Distributed, online submodular maximization algorithms that satisfy these constraints would enable decentralized network design with certifiable optimality guarantees. Currently, however, no such distributed algorithms exist.

In this paper, we study distributed submodular maximization under the computational assumptions described above. Our distributed submodular maximization approach is based on two basic insights. First, when the objective function is submodular, locally optimal solutions, in which node follows an optimal strategy given the actions of the other nodes, are within a provable bound of the global optimum [9]. Second, even when the objective function is time-varying and unknown, each node can approximate the optimal strategy over time via prediction and learning algorithms [10].

We formulate algorithms for both constrained and unconstrained distributed submodular maximization. Under our approach, a subset of nodes (e.g., a set of nodes to act as caches [2], leaders [5], or active sensors [3]) is selected by each node deciding, in a distributed fashion, whether to join the set based on the current and previous values of the objective function. In the case of unconstrained optimization, we introduce a probabilistic algorithm in which each node decides, at each tick of its internal clock, whether to join or exit the set based on the previously observed utility. We prove that, over a sufficiently long period of time, the expected utility of our algorithm is within a factor of $1/3$ of the optimum.

In the case of constrained optimization of a monotone submodular function, we propose a probabilistic algorithm in which, at each tick of its internal clock, each node decides whether or not to join the set by exchanging with a uniformly randomly chosen member of the set, based on the previously observed utility. The expected utility of our algorithm for the constrained case converges to within a factor of $1/2$ of the optimum, without requiring time synchronization, node broadcast, or global information. Through experiments, we evaluate both the unconstrained and constrained algorithms in the application of sensor selection using real-world data [11], and find that the utility achieved by our approach is within ten percent of the best centralized algorithms.

The paper is organized as follows. Section II reviews the related work. Section III states our network model and gives relevant background. Sections IV and V present our distributed algorithms for unconstrained and constrained submodular maximization, respectively. Section VI analyzes the

A. Clark, L. Bushnell, and R. Poovendran are with the Department of Electrical Engineering, University of Washington, Seattle, WA 98195 USA. awclark@uw.edu, lb2@uw.edu, rp3@uw.edu

B. Alomair is with the Center for Cybersecurity, King Abdulaziz City for Science and Technology, Riyadh, Saudi Arabia. alomair@uw.edu

978-3-901882-63-0/2014 - Copyright is with IFIP

complexity of our proposed algorithms. Section VII evaluates our approach through numerical case study. Section VIII concludes the paper and describes open problems.

II. Related Work

In the seminal work on submodular maximization with cardinality constraints, the authors derived worst-case bounds of $(1-1/e)$ for the greedy algorithm and $1/2$ for the exchange algorithm for centralized submodular maximization [9]. The bound of $(1-1/e)$ was later shown to be the best achievable unless $P = NP$ [12]. Subsequently, centralized submodular maximization algorithms with provable performance guarantees, based on convex optimization of a relaxed problem, have been proposed [8]. Unconstrained submodular maximization has also been considered, with the best known randomized algorithm achieving an expected optimality gap of $1/2$ [13].

Recently, algorithms have been proposed for submodular maximization in an online setting [14], [15]. In this case, instead of selecting a fixed set in order to maximize a given submodular objective function, a set S_t is chosen at time t without any information on the objective function f_t. The set is instead chosen based on the past objective functions $f_1, f_2, \ldots$. It has been shown that, using experts algorithms, the average optimality gap between current online submodular maximization algorithms and the utility of the best fixed set converges to $(1-1/e)$ for t sufficiently large [14].

Our approach differs from the current online submodular maximization algorithms, in that we do not assume a centralized entity with knowledge of the past objective functions. Moreover, the centralized algorithms of [14] are based on online implementation of the greedy heuristic of [9], while our algorithms are based on local exchanges, which are a fundamentally different class of algorithms.

In a submodular game, each player chooses its actions from a feasible set in a noncooperative manner in order to maximize its submodular utility function [16]. Network design problems such as multi-sensor allocation in sensor networks have been approached within the submodular game framework [17]. Our work differs from these related works in two ways. First, while the actions of a player in a submodular game are only constrained by that player's feasible set, we consider networks where the constraints on the actions of different nodes are coupled. Second, the analysis of [16], [17] assumes that each node can compute the impact of each of its actions on the objective function at each time instant. Since the computation of many common objective functions requires global knowledge of the network topology and current value of the set [2], [3], [5], we do not make this assumption.

To the best of our knowledge, the only existing work on distributed submodular maximization in either the online or offline case is in the context of distributed online sensor selection [18]. This approach is based on a stronger computational model, which assumes time synchronization and requires each node to compute the incremental benefit of being included in the set S_t, which we do not assume. Moreover, unconstrained submodular maximization is not considered in [18]. A more detailed comparison of our approach to that of [18] is presented in Section VI.

III. Network Model and Background

In this section, we state our assumptions on the computation capabilities of the distributed nodes and give needed background on multi-armed bandit prediction algorithms.

A. Network Model

We consider a set of n nodes indexed in the set $V = \{1, \ldots, n\}$. The nodes form a connected communication network in order to exchange information. The total number of nodes n is unknown to each individual node, although each node has knowledge of its one-hop neighbors. Each node is assumed to be capable of executing probabilistic polynomial-time algorithms, and can choose to join or leave the set S at any time. We let S_t denote the set at time $t \geq 0$. Each node is assumed to have an independent Poisson clock with rate 1, so that the expected number of clock ticks in $[0, T]$ is $O(T)$. The Poisson assumption is used to analyze our algorithm in Lemma 2 of Section V.

The objective function $f_t : 2^V \rightarrow \mathbb{R}_{\geq 0}$ quantifies the overall utility of the group of nodes at time t. The function f_t is assumed to be nonnegative and submodular at each time t (in Section V, we make the additional assumption that f_t is monotone); a function is submodular if for any $A \subseteq B \subseteq V$ with $v \notin B$,

$$f_t(A \cup \{v\}) - f_t(A) \geq f_t(B \cup \{v\}) - f_t(B).$$

Furthermore, we assume that there exists $K > 0$ such that $f_t(U) \leq K$ for all t and $U \subseteq V$.

Each node is assumed to have knowledge of the current value of $f_t(S_t)$ (a distributed algorithm for computing the objective function used in our numerical case study is given in Section VII). The nodes do not have knowledge of the set S_t itself, including the cardinality of S_t, or the value of $f_t(A)$ for any $A \neq S_t$. We assume that the nodes have a distributed protocol that enables any node to send a message to a uniformly random node in S_t. This can be achieved by having the nodes in S_t maintain a distributed hash table, which would require each node in S_t to store the identities of $O(\log |S_t|)$ other nodes in S_t, and would incur a routing overhead of $O(\log |S_t|)$ [19].

B. Background on Multi-Armed Bandit Algorithms

A multi-armed bandit algorithm is defined as follows [10]. Let $\mathcal{A}$ represent a set of actions. Define $\ell_1, \ell_2, \ldots,$ to be a sequence of benefit functions $\ell_j : \mathcal{A} \rightarrow \mathbb{R}_{\geq 0}$. The functions ℓ_j can be arbitrary; in particular, they can depend on the prior actions $\{a_m \in \mathcal{A} : m < j\}$.

Definition 1: A multi-armed bandit (MAB) algorithm $\mathcal{P}$ is a probabilistic algorithm that takes as input at time step j a sequence of past actions $a_1, a_2, \ldots, a_{j-1}$ and a sequence of past benefits $\ell_1(a_1), \ell_2(a_2), \ldots, \ell_{j-1}(a_{j-1})$, and outputs an action a_j.

An example of a MAB algorithm, denoted Exponential Weighted Average (EWA) [10], that will be used as a subroutine in our distributed submodular maximization approach is given as Algorithm 1. The actions chosen by EWA provide

Algorithm 1 (EWA): Prediction algorithm for computing an action a_j based on previous observed gains

```
Input: Set of actions $\mathcal{A} = \{1, \ldots, N\}$
Initialization: Set $w_{i,0} = 1$, $p_i = \frac{1}{N}$ for all $i \in \mathcal{A}$
Choose parameters $\beta$, $\eta$, $\gamma \in [0,1]$
At each round $j$:
  Select action $a_j \in \mathcal{A}$ from distribution $\mathbf{p}$, receive gain $\ell_j$
  for $i = 1, \ldots, N$
    if $a_j = i$: $\ell_i' = (\ell_j + \beta)/p_i$
    else: $\ell_i' = \beta/p_i$
    $w_i = w_i \exp(\eta \ell_i')$
  $\mathbf{p} = (\mathbf{1}^T \mathbf{w})^{-1} \mathbf{w}$
```

higher expected utility over time than choosing any fixed action at each time step, as described by the following.

Proposition 1 ([10], Theorem 6.10): When the number of actions $|\mathcal{A}| = N$, the algorithm EWA satisfies

$$\mathbf{E}\left(\sum_{j=1}^{M} \ell_j(a_j)\right) + O\left(\sqrt{(N \ln N)M}\right) \geq \max_{a \in \mathcal{A}} \left\{ \mathbf{E}\left(\sum_{j=1}^{M} \ell_j(a)\right) \right\}. \quad (1)$$

IV. Proposed Online Algorithm for Distributed Unconstrained Submodular Maximization

In this section, we present distributed algorithms for solving problems of the form

$$\text{maximize} \quad \int_0^T f_t(S_t)\ dt \quad (2)$$

If the function f_t is monotone for all t, then the solution is $S_t \equiv V$. For the non-monotone case, our approach is based on the known fact that a locally optimal solution to $\max\{f(S) : S \subseteq V\}$, i.e., a solution satisfying $f(S \cup \{a\}) \leq f(S)$ for all $a \notin S$ and $f(S - \{a\}) \leq f(S)$ for all $a \in S$, is within a provable $O(1)$ bound of the global optimum [7]. In the distributed online setting of (2), the nodes cannot verify whether local optimality holds. By choosing whether to join S_t at each epoch using the EWA algorithm, however, the nodes can guarantee convergence in expectation to a local optimum by Proposition 1.

We first define time-varying sets $A_t \subseteq V$ and $B_t \triangleq V \setminus A_t$, as well as a parameter $\delta \in [-1,1]$; we show that $\delta = 1/3$ provides an approximation factor of $1/3$. Nodes in A_t are in the set S_t with probability $p = \frac{1+\delta}{2}$, while nodes in B_t are in the set with probability $q = \frac{1-\delta}{2}$. Each node i implements the EWA algorithm discussed in Section III.

The algorithm, which we denote **Distributed-Unconstrained-Optimization (DUO)**, is as follows. At the j-th tick of its Poisson clock, node i runs the algorithm EWA, with input $(a_1^{(i)}, \ell_1), \ldots, (a_{j-1}^{(i)}, \ell_{j-1})$, where $a_m^{(i)}$ is the action taken by node i at time m, $\ell_m = \int_{T_{m-1}}^{T_m} f_t(S_t)\ dt$ and T_m is the time of the m-th clock tick. If the algorithm returns 1, then i joins S_t with probability $p = \frac{1+\delta}{2}$. If EWA returns 0, then i joins S_t with probability $q = 1 - p$.

Algorithm 2 (DUO): Algorithm for node $i \in V$ to decide membership in S_t

```
Input: Poisson clock with rate 1
Parameter $\delta \in [-1,1]$, $p = \frac{1+\delta}{2}$.
Output: Time-varying set $S_t$
At each clock tick, do
  $\ell_j \leftarrow \int_{T_{i(j-1)}}^{T_{ij}} f_t(S_t)\ dt$
  $a_j \leftarrow EWA((a_1, \ell_1), \ldots, (a_{j-1}, \ell_{j-1}))$
  if $a_j = 1$
    $S_{T_{ij}} \leftarrow \begin{cases} S_{T_{ij}} \cup \{i\} & \text{with probability } p \\ S_{T_{ij}} \setminus \{i\} & \text{with probability } (1-p) \end{cases}$
  else
    $S_{T_{ij}} \leftarrow \begin{cases} S_{T_{ij}} \cup \{i\} & \text{with probability } (1-p) \\ S_{T_{ij}} \setminus \{i\} & \text{with probability } p \end{cases}$
  end
end
```

The following results, which are proved in the appendix, describe the local optimality of DUO.

Proposition 2: For any $\delta \in [-1,1]$, the sets $\{S_t : t \in [0,T]\}$ chosen by DUO satisfy

$$\begin{aligned} &\mathbf{E}\int_0^T f_t(S_t)\ dt + O(\sqrt{2T\ln 2}) \\ &\geq \max\left\{\mathbf{E}\int_0^T f_t(R(A_t - \{i\}, \delta))\ dt, \right. \\ &\qquad \left. \mathbf{E}\int_0^T f_t(R(A_t \cup \{i\}, \delta))\ dt\right\} \end{aligned}$$

for each $i \in V$.

The local optimality of the set S_t follows as a corollary to Proposition 2.

Corollary 1: Let $v \in V$, and define $\mathcal{T} := \{t \in [0,T] : v \in A_t\}$. For any $v \in V$, $\delta \in [-1,1]$, the sets $\{S_t : t \in [0,T]\}$ chosen by DUO satisfy

$$\begin{aligned} &\max\left\{\mathbf{E}\int_{\mathcal{T}^c} f_t(S_t \cup \{v\}) - f_t(S_t - \{v\})\ dt, \right. \\ &\left. \mathbf{E}\int_{\mathcal{T}} f_t(S_t - \{v\}) - f_t(S_t \cup \{v\})\ dt\right\} \leq O\left(\frac{\sqrt{2T\ln 2}}{\delta}\right). \end{aligned} \quad (3)$$

Based on the local optimality of our approach, the overall optimality gap of DUO is given as follows. The proof is omitted due to space constraints.

Theorem 1: Let $C = \arg\max\{\int_0^T f_t(S)\ dt : S \subseteq V\}$. When $\delta = \frac{1}{3}$, the sets $\{S_t : t \in [0,T]\}$ selected by DUO satisfy

$$\mathbf{E}\int_0^T f_t(S_t)\ dt + O(3n\sqrt{2T\ln 2}) \geq \frac{1}{3}\int_0^T f_t(C)\ dt.$$

Theorem 1 implies that the algorithm DUO achieves an $O(1)$ optimality gap in expectation, for the case where there is no constraint on the set S_t. The case where there is a constraint on the size of S_t is discussed in the following section.

V. Proposed Online Algorithm for Distributed Constrained Submodular Maximization

In this section, we present our distributed online algorithm for approximating the solution to

$$\begin{array}{ll} \text{maximize} & \int_0^T f_t(S_t)\ dt \\ \text{s.t.} & |S_t| \leq k \end{array} \qquad (4)$$

which represents maximization of the time-varying submodular functions $f_t(S_t)$ subject to a cardinality constraint. We make the additional assumption that f_t is nondecreasing for all t, i.e., that $f_t(A) \leq f_t(B)$ for all $A \subseteq B$.

Our algorithm is a distributed implementation of the exchange heuristic [9]. In the offline, centralized version of the algorithm, if there exist nodes $u \in V \setminus S$ and $v \in S$ with $f(S) \leq f(S \cup \{u\} - \{v\})$, then the set S is updated to $S \cup \{u\} - \{v\}$ and we say that nodes u and v are exchanged. In our distributed version, at each tick of its internal Poisson clock, each node u probabilistically decides, using the EWA algorithm, whether to exchange with a node in S_t. If so, node u chooses a random node $v \in S_t$ using a distributed protocol and requests an exchange. Node v decides whether to accept the exchange using the EWA algorithm, and if v chooses to accept, then the set is updated to $S_t \cup \{u\} - \{v\}$.

In what follows, we first describe the algorithm, denoted **Distributed-Constrained-Optimization (DCO)**, used by each node to perform the decentralized optimization. We then analyze the optimality gap. Proofs are in the appendix.

At each tick of its Poisson clock, node u chooses an action based on whether $u \in S_t$, where t is the current time. If $u \in S_t$, then u takes no further action. If $u \notin S_t$, then u decides to either join S_t or remain in $V \setminus S_t$. In the case where join is chosen, node u chooses a node v uniformly at random from S_t, as discussed in Section III-A, and requests an exchange. If v accepts, then S_t is updated to $S_t - \{v\} \cup \{u\}$. Otherwise, S_t remains unchanged.

Nodes decide whether to join S_t, and whether to accept an exchange, based on EWA. The four actions at each time are $(join, accept)$, $(join, refuse)$, $(stay, accept)$, and $(stay, refuse)$. If $u \notin S_t$ and the action chosen by the algorithm is $(join, accept)$ or $(join, refuse)$, then u will attempt to join S_t. Otherwise, u will not attempt to join. If $u \in S_t$ and another node requests an exchange, then u will accept if the algorithm outputs $(join, accept)$ or $(stay, accept)$, and refuse otherwise. If a node makes a decision at time T and the next decision is at time T', then the payoff for the decision at time T is given by $\int_T^{T'} f_t(S_t)\ dt$.

The structure of the optimality gap analysis is as follows. We first prove a sequence of lemmas, which imply that

$$\int_0^T f_t(S_t)\ dt + o(T) \geq \int_0^T \frac{1}{k} \sum_{v \in S_t} f_t(S_t - \{v\} \cup \{u\}) \quad (5)$$

for all $u \in V$. We then use this result to prove that, for any fixed set C with $|C| \leq k$,

$$\int_0^T f_t(S_t)\ dt + o(T) \geq \frac{1}{2} \int_0^T f_t(C)\ dt.$$

In order to prove (5), we divide the set S_t into k slots. At each time t, slot m contains one node from S_t, denoted v_t^m, with $v_t^m \neq v_t^{m'}$ for $m \neq m'$, and $S_t = \bigcup_{m=1}^k \{v_t^m\}$. Initially, the mapping between nodes and slots is arbitrary. If an exchange takes place at time t, with node u joining S_t and v exiting S_t, then v_t^m is updated to u, where m satisfies $v_t^m = v$. Using this definition, we have the following lemma.

Algorithm 3 (DCO): Algorithm for node u to decide membership in S_t under cardinality constraint

```
Input: Poisson clock with rate 1
Output: Time-varying set $S_t$
On each clock tick, do:
  if $u \notin S_t$
    Update EWA with gain $\int_{T_{j-1}}^{T_j} f_t(S_t)\ dt$
    if EWA outputs $join$
      Send exchange request to randomly chosen $v \in S_t$
      if $v$ accepts exchange: $S_t = S_t - \{v\} \cup \{u\}$
end
if $u \in S_t$ and receive exchange request from $u' \notin S_t$
  Update EWA with gain $\int_{T_{j-1}}^{T_j} f_t(S_t)\ dt$
  if EWA outputs $accept$: $S_t = S_t - \{u\} \cup \{u'\}$
end
```

Lemma 1: For any $u \in V$ and $m \in \{1, \ldots, k\}$, define $T_m \triangleq \min\{t : v_t^m = u\}$. The sets $\{S_t : t \in [0, T]\}$ chosen by DCO satisfy

$$\mathbf{E} \int_{T_m}^T f_t(S_t)\ dt + O(\sqrt{4T \ln 4}) \geq \mathbf{E} \int_{T_m}^T f_t(S_t - v_t^m + u)\ dt. \quad (6)$$

Lemma 1 implies that, for any node, the algorithm DCO provides higher overall utility than occupying slot m in the interval $[T_m, T]$. We next show that the result of Lemma 1 can be extended to $[0, T]$, due to the fact that $\mathbf{E}(T_m) < \infty$.

Lemma 2: For any $u \in V$, $m \in \{1, \ldots, k\}$, the sets $\{S_t : t \in [0, T]\}$ chosen by DCO satisfy

$$\mathbf{E} \int_0^T f_t(S_t)\ dt + O(\sqrt{4T \ln 4}) \geq \mathbf{E} \int_0^T f_t(S_t - \{v_t^m\} \cup \{u\})\ dt.$$

Lemmas 1 and 2 imply that, for each node, following DCO provides higher utility than remaining in any slot m for the entire interval $[0, T]$. Summing over the slots $m = 1, \ldots, k$ yields (5), as shown in the following proposition.

Proposition 3: For any $u \in V$, the sets $\{S_t : t \in [0, T]\}$ chosen by DCO satisfy

$$\mathbf{E} \int_0^T f_t(S_t)\ dt + O(\sqrt{4T \ln 4}) \geq \frac{1}{k} \mathbf{E} \int_0^T \sum_{v \in S_t} f_t(S_t - \{v\} \cup \{u\})\ dt. \quad (7)$$

By Proposition 3, the utility of the set sequence S_t chosen by DCO cannot be improved by replacing a randomly chosen node in S_t with any fixed node u. The following theorem uses this result to derive the optimality gap of DCO.

Theorem 2: The sets $\{S_t : t \in [0,T]\}$ chosen by DCO satisfy

$$\mathbf{E}\int_0^T f_t(S_t)\ dt + O(k\sqrt{4T\ln 4}) \geq \frac{1}{2}\int_0^T f_t(C)\ dt$$

for all C with $|C| \leq k$.

Theorem 2 shows that, in expectation, algorithm DCO achieves the same optimality gap as the offline exchange heuristic [9]. This is in spite of the fact that, while the offline exchange heuristic is based on identifying a specific pair of elements u and v satisfying $f(S - \{u\} \cup \{v\}) \geq f(S)$, DCO operates by performing a sequence of random exchanges. The use of random exchanges therefore reduces the memory, computation, and information requirements of each node, without sacrificing worst-case optimality guarantees.

VI. Performance Evaluation and Comparison

In what follows, we evaluate our proposed algorithms for constrained and unconstrained submodular maximization, and compare with the constrained submodular maximization algorithm **Distributed-online-greedy** (DOG) introduced in [18]. We consider the following criteria: (a) the computational assumptions of the nodes specified by the algorithms, (b) the asymptotic bounds that are achieved, (c) the rate at which the algorithms must be computed in order to reach a given value ϵ from the asymptotic bound, (d) the communication overhead, and (e) the storage overhead at each node.

Our analysis makes the same assumptions as that of [18] in order to guarantee consistency of the results, with one exception. While [18] assumed that the cost of broadcasting a message to all other nodes is 1, we assume that nodes communicate via multi-hop paths, leading to a cost of broadcast of $O(\frac{n}{\pi} - \sqrt{n})$ [20]. Similarly, we assume that the cost of sending a message to a single node is proportional to the network diameter, i.e., $O(\sqrt{n})$. The comparison results are summarized below in Table I.

TABLE I
Performance Comparison of Distributed Submodular Maximization Algorithms

Criterion	**DOG** [18]	**DCO**	**DUO**
Constraints	Cardinality k	Cardinality k	None
Optimality Gap	$(1-1/e) \approx 0.63$	0.5	0.33
Communication	$O\left(kn^2 \log n\right)$	$O(2n\sqrt{n})$	0
Storage	$O(k)$	$O(1)$	$O(1)$

VII. Case Study: Distributed Sensor Scheduling

In this section, we evaluate our approach through a case study, based on the problem of determining the optimal set of nodes to activate in a sensor network. We first provide the details of our experimental setup. We then describe our approach for distributed computation of the objective function, followed by the simulation results.

A. Experimental Setup

We evaluate our proposed algorithms using Matlab. In this study, we consider the problem of deciding which sensors, out of a set of $n = 46$ sensors, should be active at a given time. To perform the study, we use the temperature monitoring data from the Intel Research Lab [11], which contains a time series of temperature measurements from a network of 46 sensors. We assumed that each temperature measurement corresponded to a sample from an underlying Gaussian random distribution μ_i corresponding to the node i. We then computed the covariance matrix K for the sensors. We let the utility function $f(S)$, which we assume to be fixed, be given as $f(S) = \frac{MMSE(\emptyset) - MMSE(S)}{MMSE(\emptyset)}$, where $MMSE(S)$ denotes the conditional variance of the measurements of each sensor, given knowledge of the measurements at locations S. The analysis considered both unconstrained and cardinality-constrained distributed optimization. For the EWA algorithm, we chose $\gamma = 0.01$, $\eta = 1/MMSE(\emptyset)$, and $\beta = 0.03$, based on the discussion of parameter selection in [10].

B. Sensing Model and Objective Function Definition

We assume that each sensor node $v \in V$ is capable of measuring a random variable X_v. The variables $\mathbf{X} \triangleq \{X_v : v \in V\}$ are jointly Gaussian with mean 0 and covariance matrix Σ. The precision matrix Q is defined by $Q \triangleq \Sigma^{-1}$. The objective function $f(S_t)$ can be written as

$$f(S_t) \triangleq \sum_{v \in V \setminus S_t} \text{var}(X_v | X_{S_t}), \tag{8}$$

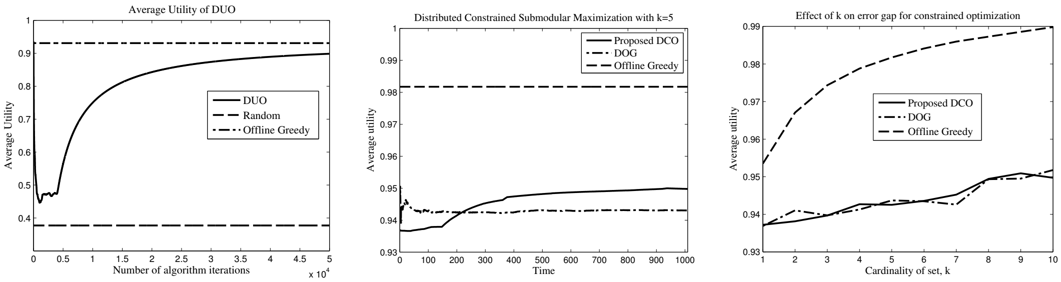
where $X_{S_t} \triangleq \{X_v : v \in S_t\}$ and

$$\text{var}(X_v|X_{S_t}) = \int_{x \in \mathbb{R}^{|S_t|}} \text{var}(X_v | X_{S_t} = x) p_{X_{S_t}}(x)\ dx.$$

This objective function measures the uncertainty of estimating the measurements at the inactive sensors, given the measurements at the active sensors. The function $f(S_t)$ was shown to be submodular as a function of S_t in [21].

To compute the objective function, a random process $z_v[m]$, $m = 1, 2, \ldots$, is simulated at each node $v \in V\ S_t$, such that the variance of $z_v[m]$ converges to $(Q_{S_t}^{-1})_{vv}$. The process is created by having each node maintain a time-varying state, which is equal to a linear combination of the states of its neighbors plus additional Gaussian noise. Each node can then estimate $\text{var}(X_v|X_{S_t})$ by computing the sample variance.

This approach assumes that nodes are capable of local synchronization, in which each node $v \in V \setminus S_t$ knows which messages to expect before its m-th iteration (in this case, the states $\{z_u[m-1] : u \in N(v) \setminus S_t\}$) [22, Ch. 1.4]. Local synchronization requires less overhead than global synchronization, which requires all nodes to agree on a common time index, and hence is a common assumption in sensor networks [23]. Furthermore, nodes in S_t do not send state values at any iteration, and are therefore not incorporated in state updates. Hence, each node $v \in V \setminus S_t$ does not require any knowledge of S_t, since these nodes are automatically excluded from computation of the algorithm.



(a) (b) (c)

Fig. 1. Numerical evaluation of our proposed algorithms DUO and DCO for distributed online submodular maximization and comparison with DOG algorithm [18] on a network of 46 nodes [11]. The reduction in mean square error is used as a utility function. (a) Utility as a function of time for the case of unconstrained optimization. After an initial learning phase, the utility approaches the maximum possible value of 1, consistently outperforming random sensor selection. (b) Performance in the case where only $k = 5$ sensors can be active. Performance of the algorithm DCO is within 0.07 of the best achievable value of 1, and is close to the centralized and distributed greedy algorithms. (c) Increase in utility as a function of the number of sensors k. The increase from the DCO and DOG algorithms is roughly linear, while the increase of the centralized algorithm is concave.

A Lyapunov function argument shows that the variance σ_v^2 of node v converges to $\text{var}(X_v|X_{S_t})$; we omit the proof due to space constraints. It remains for each node to compute the average variance $f(S_t)$ over all the nodes $u \notin S_t$.

C. Experimental Results

For the unconstrained case, the overall utility function was equal to $f(S) - c|S|$, where c represents the cost of activating each sensor; we chose $c = \frac{1}{n}$, so that the total cost is equal to the fraction of sensors activated. For comparison, we also evaluated the performance of random sensor activation. As shown in Figure 1(a), the distributed online approach first experiences a rapid decrease in utility, as multiple nodes randomly decide to activate, thus increasing the cost. As the time progresses, these nodes observe the decrease in their utility and deactivate, eventually approaching a utility of 0.9. Since 1 is an upper bound on the maximum achievable utility, DUO is at least 0.9-optimal for this problem. At all time periods, the algorithm DUO provides higher utility than random selection, and converges to a comparable utility to the offline greedy algorithm.

Figure 1(b) shows the utility of the constrained case as a function of time with $k = 5$. The algorithm DCO converges rapidly to a utility of 0.945. As a comparison, a centralized greedy algorithm provides utility of 0.97, while 1 is an upper bound on the utility that can be achieved. Moreover, DCO provides higher utility than the DOG algorithm. The average utility as a function of the constraint k is shown in Figure 1(c). Clearly, increasing the number of sensors that can be activated increases the utility. We observe a concave increase in utility for centralized greedy approach and a roughly linear increase for DCO and DOG. The rate of increase for the distributed algorithm is lower than that of the centralized case; however, when $k = 10$, DCO gives utility of 0.95 while the greedy algorithm gives 0.99.

VIII. Conclusions and Open Problem

We studied the problem of submodular maximization in a distributed, online setting, in which each node must decide whether to join a given set in order to maximize a submodular objective function. We considered two versions of the distributed submodular maximization problem. In the first case, there are no constraints on which nodes can join the set. In the second case, the number of nodes in the set cannot exceed a fixed number k and the function to be maximized is assumed to be monotone.

For both cases, we introduced distributed algorithms for submodular maximization with $O(1)$ expected optimality gap. In the unconstrained case, our distributed algorithm achieves a 0.33 optimality gap, compared to the best centralized algorithm's 0.5 optimality gap. Our algorithm for maximization in the constrained case achieves a 0.5 optimality gap, compared to a (1-1/e) gap for centralized algorithms. Both algorithms can be computed by nodes with limited computation, communication, and storage capabilities, and without knowledge of the number of nodes, any cardinality constraint, or access to a value oracle of the objective function. An experimental study using sensor network data showed that both algorithms provide near-optimal empirical performance.

The optimality guarantee of $1/2$ achieved by our proposed algorithm DCO is less than the $(1 - 1/e)$ guarantee achieved by the greedy centralized algorithm. In our future work, we will investigate additional problem structure that may enhance the performance of local search algorithms such as DCO. One possible set of problems arise in facility location, where exchange-based algorithms are known to provide near-optimal solutions [24].

Acknowledgment

This work was supported by a grant from the King Abdulaziz City for Science and Technology.

References

[1] M. Andrews and L. Zhang, "Scheduling algorithms for multi-carrier wireless data systems," *Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, pp. 3–14, 2007.

[2] N. Golrezaei, K. Shanmugam, A. G. Dimakis, A. F. Molisch, and G. Caire, "Femtocaching: Wireless video content delivery through distributed caching helpers," *Proceedings of IEEE Infocom*, pp. 1107–1115, 2012.

[3] A. Krause, R. Rajagopal, A. Gupta, and C. Guestrin, "Simultaneous optimization of sensor placements and balanced schedules," *IEEE Transactions on Automatic Control*, vol. 56, no. 10, pp. 2390–2405, 2011.

[4] Z. Zheng and N. B. Shroff, "Maximizing a submodular utility for deadline constrained data collection in sensor networks," *10th IEEE International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*, pp. 116–123, 2012.

[5] A. Clark, L. Bushnell, and R. Poovendran, "Leader selection for minimizing convergence error in leader-follower systems: A supermodular optimization approach," *10th IEEE International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*, pp. 111–115, 2012.

[6] S. Fujishige, *Submodular functions and optimization*. Elsevier Science, 2005, vol. 58.

[7] U. Feige, V. Mirrokni, and J. Vondrak, "Maximizing non-monotone submodular functions," *SIAM Journal on Computing*, vol. 40, no. 4, pp. 1133–1153, 2011.

[8] G. Calinescu, C. Chekuri, M. Pal, and J. Vondrak, "Maximizing a submodular set function subject to a matroid constraint," *SIAM Journal on Computing*, vol. 40, no. 6, pp. 1740–1766, 2011.

[9] G. Nemhauser, L. Wolsey, and M. Fisher, "An analysis of approximations for maximizing submodular set functions - I," *Mathematical Programming*, vol. 14, no. 1, pp. 265–294, 1978.

[10] N. Cesa-Bianchi and G. Lugosi, *Prediction, Learning, and Games*. Cambridge University Press, 2006.

[11] "Intel Berkeley Sensor Network Dataset," http://db.csail.mit.edu/labdata/labdata.html.

[12] U. Feige, "A threshold of ln n for approximating set cover," *Journal of the ACM (JACM)*, vol. 45, no. 4, pp. 634–652, 1998.

[13] N. Buchbinder, M. Feldman, J. S. Naor, and R. Schwartz, "A tight linear time (1/2)-approximation for unconstrained submodular maximization," *IEEE 53rd Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 649–658, 2012.

[14] M. Streeter and D. Golovin, "An online algorithm for maximizing submodular functions," *Advances in Neural Information Processing Systems (NIPS)*, pp. 1577–1584, 2008.

[15] V. Gabillon, B. Kveton, Z. Wen, B. Eriksson, and S. Muthukrishnan, "Adaptive submodular maximization in bandit setting," *Advances in Neural Information Processing Systems*, pp. 2697–2705, 2013.

[16] E. Altman and Z. Altman, "S-modular games and power control in wireless networks," *IEEE Transactions on Automatic Control*, vol. 48, no. 5, pp. 839–842, 2003.

[17] C. Wu, Y. Xu, Y. Chen, and C. Lu, "Submodular game for distributed application allocation in shared sensor networks," *Proceedings of IEEE Infocom*, pp. 127–135, 2012.

[18] D. Golovin, M. Faulkner, and A. Krause, "Online distributed sensor selection," *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, pp. 220–231, 2010.

[19] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," *ACM SIGCOMM Computer Communication Review*, vol. 31, no. 4, pp. 149–160, 2001.

[20] T. Calamoneri, A. Clementi, M. Ianni, M. Lauria, A. Monti, and R. Silvestri, "Minimum energy broadcast and disk cover in grid wireless networks," *Theoretical Computer Science*, vol. 399, pp. 38–53, 2008.

[21] 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, pp. 235–284, 2008.

[22] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation*. Old Tappan, NJ (USA); Prentice Hall Inc., 1989.

[23] P. Ogren, E. Fiorelli, and N. E. Leonard, "Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment," *IEEE Transactions on Automatic Control*, vol. 49, no. 8, pp. 1292–1302, 2004.

[24] V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, and V. Pandit, "Local search heuristics for k-median and facility location problems," *SIAM Journal on Computing*, vol. 33, no. 3, pp. 544–562, 2004.

Appendix

Proof of Proposition 2: Suppose that each node follows the DUO algorithm. Let S_t denote the set at time t, and define $\hat{S}_t = S_t \setminus \{i\}$ (note that $S_t = \hat{S}_t$ if $i \notin S_t$). Furthermore, suppose that the Poisson clock ticks of node i occur at times $(T_1, \ldots, T_M)$. Let $\hat{a}_j$ denote the action of node i at the j-th clock tick.

Define a related multi-armed bandit problem with M time steps as follows. The action space $\mathcal{A} = \{0, 1\}$, while the gain $\hat{\ell}_j$ from action $a_j \in \mathcal{A}$ at step j is given by

$$\hat{\ell}_j = \left\{ \begin{array}{ll} \int_{T_j}^{T_{j+1}} f_t(\hat{S}_t \cup \{i\})\ dt, & a_j = 1 \\ \int_{T_j}^{T_{j+1}} f_t(\hat{S}_t)\ dt, & a_j = 0 \end{array} \right.$$

Suppose that the EWA algorithm is run with this action space and set of gains, using the same parameters as in the subroutine of DUO, outputting actions $(a_1^*, \ldots, a_M^*)$. First, note that the expected gain of the algorithm is

$$\sum_{j=1}^{M} \ell_j(a_j = a_j^*) = \mathbf{E} \int_0^T f_t(S_t)\ dt, \tag{9}$$

since a_j^* and $\hat{a}_j$ are obtained by running the same randomized algorithm with the same parameters and inputs. Second, note that if $a_j \equiv 1$ for $j = 1, \ldots, M$, then

$$\begin{aligned} \sum_{j=1}^{M} \ell_j(a_j = 1) &= \sum_{j=1}^{M} \int_{T_j}^{T_{j+1}} f_t(\hat{S}_t \cup \{i\})\ dt \\ &= \int_0^T f_t(S_t \cup \{i\})\ dt. \end{aligned} \tag{10}$$

Similarly, if $a_j \equiv 0$ for $j = 1, \ldots, M$, then

$$\begin{aligned} \sum_{j=1}^{M} \ell_j(a_j = 0) &= \sum_{j=1}^{M} \int_{T_j}^{T_{j+1}} f_t(\hat{S}_t)\ dt \\ &= \int_0^T f_t(S_t \setminus \{i\})\ dt. \end{aligned} \tag{11}$$

Combining (9), (10), and (11) and applying Proposition 1 yields the desired result. ■

Proof of Corollary 1: By Proposition 2, we have

$$\begin{aligned} O(\sqrt{2T \ln 2}) \geq \mathbf{E} &\int_{\{t : v \notin A_t\}} f_t(R(A_t \cup \{v\}, \delta)) \\ &\quad - f_t(R(A_t, \delta))\ dt \\ = \mathbf{E} &\int_{\{t : v \notin A_t\}} [p f_t(S_t \cup \{v\}) + q f_t(S_t - \{v\}) \\ &\quad - (q f_t(S_t \cup \{v\}) + p f_t(S_t - \{v\}))]\ dt \\ = \delta \mathbf{E} &\int_{\{t : v \notin A_t\}} f_t(S_t \cup \{v\}) - f_t(S_t - \{v\})\ dt. \end{aligned}$$

The proof of the second part of Corollary 1 is similar. ■

Proof of Lemma 1: Consider a multi-armed bandit problem from the perspective of node u. Let $\hat{I}_t$ be a process with actions $(join, accept)$, $(join, refuse)$, $(stay, accept)$, and $(stay, refuse)$, which receives payoffs $\int_T^{T'} f_t(S_t)\ dt$. The outcome of the MAB problem is the random set sequence $\{\hat{S}_t^u : t \in [T, T']\}$, which denotes the set chosen by the nodes other than u based on the actions $\hat{I}_t$, along with the current state of the node u (the current slot of node u).

If u is in slot 0 and a $join$ action is chosen, then the payoff is $\mathbf{E} \int_T^{T'} f_t(\hat{S}_t^u \cup \{u\})\ dt$. If $stay$ is chosen, then the payoff is $\mathbf{E} \int_T^{T'} f_t(\hat{S}_t^u)\ dt$. If u is in slot m and accepts an exchange, then the payoff is $\mathbf{E} \int_T^{T'} f_t(\hat{S}_t^u)\ dt$. Finally, if u is in slot m and refuses the exchange, then the payoff is $\mathbf{E} \int_T^{T'} f_t(\hat{S}_t^u - \{v_t^m\} \cup \{u\})$.

Now, if at each iteration, node u follows EWA, then the total expected payoff is equivalent to the payoff from choosing $\hat{I}_t$ at each iteration, which is $\mathbf{E}\int_{T_m}^{T} f_t(S_t)\ dt$. On the other hand, if node u chooses $(join, refuse)$ at each time slot, then the expected payoff is $\mathbf{E}\int_{T_m}^{T} f_t(S_t - \{v_m^t\} \cup \{u\})\ dt$. Proposition 1 and the fact that there are 4 possible actions at each time step complete the proof. ■

Proof of Lemma 2: By Lemma 1,

$$\mathbf{E}\int_{T_m}^{T} f_t(S_t)\ dt + o(T) \geq \mathbf{E}\int_{T_m}^{T} f_t(S_t - v_t^m \cup \{u\})\ dt.$$

Hence, it suffices to show that

$$\mathbf{E}\int_{0}^{T_m} f_t(S_t)\ dt + o(T) \geq \mathbf{E}\int_{0}^{T_m} f_t(S_t - v_t^m \cup \{u\})\ dt,$$

which is equivalent to

$$\mathbf{E}\int_{0}^{T_m} (f_t(S_t - v_t^m \cup \{u\}) - f_t(S_t))\ dt \leq o(T).$$

Since $f_t(S) \leq K$ for all t and S,

$$\mathbf{E}\int_{0}^{T_m} (f_t(S_t - v_t^m \cup \{u\}) - f_t(S_t))\ dt \leq \mathbf{E}\int_{0}^{T_m} K\ dt = K\mathbf{E}(T_m)$$

and it therefore suffices to show that $\mathbf{E}(T_m) = o(T)$.

Let $\hat{T}_1, \ldots, \hat{T}_r, \ldots$ denote the sequence of times when a clock tick occurs and $u \notin S_t$. Let $X = \min\{r : v_{\hat{T}_r}^m = u\}$. Hence $\mathbf{E}(T_m) = \mathbf{E}(\hat{T}_X)$. Let A_i denote the event that node u chooses $join$ at time $\hat{T}_i$.

This expectation can be written as the summation $\sum_{i=1}^{X} \mathbf{E}(\hat{T}_i - \hat{T}_{i-1})$. We have that $\mathbf{E}(\hat{T}_i - \hat{T}_{i-1}) = \mathbf{E}(\hat{T}_i - \hat{T}_{i-1}|A_i)Pr(A_i) + \mathbf{E}(\hat{T}_i - \hat{T}_{i-1}|A_i^c)Pr(A_i^c)$.

If A_i holds, then the time before the next decision to $join$ or $stay$ is equal to the time for another node to initiate an exchange and for node u to accept. Each node chooses to $join$ or $stay$ according to an independent Poisson process with unit rate. At each tick, the probability of choosing $join$ is (by definition of algorithm EWA) bounded below by $\frac{\gamma}{2}$. The probability of initiating an exchange with node u is $\frac{1}{k}$, and so exchange requests from each other node arrive according to a Poisson process with rate bounded below by $\frac{2}{k\gamma}$. Furthermore, since each node has an independent Poisson clock, exchange requests arrive at u according to a Poisson process with rate bounded below by $\frac{2(n-k)}{k\gamma}$. By definition of algorithm EWA, the probability that node u accepts each request is at least $\frac{\gamma}{2}$. Thus $\mathbf{E}(\hat{T}_i - \hat{T}_{i-1}|A_i) \leq \frac{4(n-k)}{\gamma^2 k} + 1$, where the 1 is the expected time to wait for the next clock tick after leaving S_t.

If A_i does not hold, the expected time for the next clock tick is 1. Hence $\mathbf{E}(\hat{T}_i - \hat{T}_{i-1}) \leq 1 + \frac{4(n-k)}{\gamma^2 k}$, and

$$\mathbf{E}(\hat{T}_X) \leq \mathbf{E}\sum_{i=1}^{X}\left(1 + \frac{4(n-k)}{\gamma^2 k}\right) = \mathbf{E}(X)\left(1 + \frac{4(n-k)}{\gamma^2 k}\right),$$

which is bounded above by $\frac{4k}{\gamma^2} + \frac{16(n-k)}{\gamma^4 k}$. Since this bound is independent of T, the $O(\sqrt{4T\log 4})$ term dominates asymptotically. ■

Proof of Proposition 3: By Lemma 2, for each slot m,

$$\mathbf{E}\int_{0}^{T} f_t(S_t)\ dt + O(\sqrt{4T\log T}) \geq \mathbf{E}\int_{0}^{T} f_t(S_t - v_t^m \cup \{u\})\ dt.$$

Summing both sides over m yields

$$\begin{aligned} &k\left(\mathbf{E}\int_{0}^{T} f_t(S_t)\ dt + O(\sqrt{4T\log 4})\right) \\ &\geq \mathbf{E}\sum_{m=1}^{k}\int_{0}^{T} f_t(S_t - v_t^m \cup \{u\})\ dt \\ &= \mathbf{E}\int_{0}^{T}\sum_{v \in S_t} f_t(S_t - \{v\} \cup \{u\})\ dt. \end{aligned}$$

Dividing by k gives the desired result. ■

Proof of Theorem 2: Monotonicity of f_t implies that

$$\mathbf{E}\int_{0}^{T} f_t(C) - f_t(S_t)\ dt \leq \mathbf{E}\int_{0}^{T} f_t(C \cup S_t) - f_t(S_t)\ dt.$$

Define a sequence of sets $U_0^t \subseteq U_1^t \subseteq \cdots \subseteq U_k^t$, with $U_0^t = S_t$, $U_k^t = S_t \cup C$, and $U_j^t - U_{j-1}^t = \{a_i^t\}$ for some $a_i^t \in C$. Then we have

$$\begin{aligned} &\mathbf{E}\int_{0}^{T} f_t(C) - f_t(S_t)\ dt && (12) \\ &\leq \mathbf{E}\int_{0}^{T}\sum_{j=1}^{k}(f_t(U_j^t) - f_t(U_{j-1}^t)) \\ &= \mathbf{E}\int_{0}^{T}\sum_{j=1}^{k}(f_t(U_{j-1}^t \cup \{a_j^t\}) - f_t(U_{j-1}^t)) \\ &\leq \mathbf{E}\int_{0}^{T}\sum_{j=1}^{k}(f_t(S_t \cup \{a_j^t\}) - f_t(S_t)) && (13) \\ &= \mathbf{E}\int_{0}^{T}\sum_{a \in C}(f_t(S_t \cup \{a\}) - f_t(S_t)) \\ &\leq \int_{0}^{T}\left[\sum_{a \in C}\mathbf{E}\left(\frac{1}{k}\sum_{v \in S_t} f_t(S_t \cup \{a\} - \{v\})\right.\right. \\ &\qquad \left.\left. - f_t(S_t - \{v\})\right)\right], && (14) \end{aligned}$$

where (13) and (14) follow from submodularity of f_t. We now use Proposition 3 to upper bound (14) as

$$\begin{aligned} &\mathbf{E}\int_{0}^{T} f_t(C) - f_t(S_t)\ dt && (15) \\ &\leq \int_{0}^{T}\sum_{a \in C}\mathbf{E}\left(\frac{1}{k}\sum_{v \in S_t} f_t(S_t) - f_t(S_t - \{v\})\right) dt && (16) \\ &\quad + O(k\sqrt{4T\log 4}) && (17) \\ &\leq \mathbf{E}\int_{0}^{T} f_t(S_t)\ dt + O(k\sqrt{4T\log 4}), && (18) \end{aligned}$$

where (18) follows from submodularity of f_t. The result follows after rearranging terms. ■