

Distributed Maximization of Submodular and Approximately Submodular Functions

Lintao Ye and Shreyas Sundaram

Abstract—We study the problem of maximizing a submodular function, subject to a cardinality constraint, with a set of agents communicating over a connected graph. We propose a distributed greedy algorithm that allows all the agents to converge to a near-optimal solution to the global maximization problem using only local information and communication with neighbors in the graph. The near-optimal solution approaches the $(1 - 1/e)$ approximation of the optimal solution to the global maximization problem with an additive factor that depends on the number of communication steps in the algorithm. We then analyze convergence guarantees of the proposed algorithm. This analysis reveals a tradeoff between the number of communication steps and the performance of the algorithm. Finally, we extend our analysis to nonsubmodular settings, using the notion of approximate submodularity.

I. INTRODUCTION

In recent years, the analysis of large-scale networks has received much attention from researchers, where the networks consist of a group of agents with different local objective functions. For such networks, the goal is to design a resource allocation method that operates in a decentralized way with local communication and fast convergence to an (approximately) optimal operating point. Scenarios where the local objective functions depend on the entire resource allocation vector are of particular interest. For such scenarios, there is a vast literature on designing distributed algorithms that guarantee convergence of the solution obtained by each agent to an optimizer of the average of all the local objective functions (e.g., [1]–[5] and the references therein).

Much of the existing work has been devoted to optimization problems in the continuous domain, where the local objective functions are convex. In contrast, settings with (discrete) submodular objective functions have been less explored (e.g., [6]–[8]). Nonetheless, the problem of maximizing submodular functions (subject to constraints) arises in many different applications including, for instance, budget allocation [9], sensor placement [10], sensor scheduling [11] and influence maximization in social networks [12]. Thus, in this paper we focus on scenarios in distributed optimization where the local objective functions are submodular.

Related Work

In [6], the authors considered maximizing a discrete submodular function subject to a general matroid constraint and proposed a decentralized algorithm to solve this problem. The algorithm relies on first lifting the local discrete submodular functions to continuous domains and then applying

This research was supported by NSF grant CMMI-1635014. Lintao Ye and Shreyas Sundaram are with the School of Electrical and Computer Engineering at Purdue University. Email: {ye159,sundara2}@purdue.edu.

appropriate rounding schemes to the obtained solution. The authors in [7] considered the scenario where a group of agents sequentially maximize a submodular function. The problem reduces to the canonical problem of maximizing a submodular function subject to a partitioned matroid constraint, which can be solved by greedy algorithms with a $1/2$ multiplicative approximation ratio [13]. In contrast, we consider the setting where a group of agents maximize a (global) submodular function (subject to a cardinality constraint) in parallel using a decentralized (i.e., distributed) greedy algorithm, which does not require any lifting or rounding process.

Regarding the maximization of nonsubmodular functions, the notion of approximate submodularity has been used to provide performance guarantees for (centralized) greedy algorithms applied to such problems (e.g., [14]–[16]). Here, we aim to propose a distributed greedy algorithm that can also solve such problems in a distributed manner.

Contributions

We propose a distributed greedy algorithm with a group of agents communicating over a connected network, which allows each agent to converge to within an additive factor of the $(1 - 1/e)$ approximation of the optimal solution to the global optimization problem. This additive factor is a function of the number of the agents, the cardinality constraint, properties of the local functions and design parameters in the distributed greedy algorithm. In particular, the analysis reveals a tradeoff between the performance of the algorithm and the number of communication steps in the algorithm. Finally, we extend our analysis to cases when the objective function is approximately submodular.

Notation and terminology

The sets of integers and real numbers are denoted as \mathbb{Z} and \mathbb{R} , respectively. For $x \in \mathbb{R}$, let $|x|$ denote its absolute value. For a set \mathcal{S} , let $|\mathcal{S}|$ denote its cardinality. Let $\mathbf{1}_n$ denote a column vector of dimension n with all of its elements equal to 1. For a matrix $P \in \mathbb{R}^{m \times n}$ and a vector $y \in \mathbb{R}^n$, let P' and y' be their transposes, respectively. Let P_{ij} denote the element in the i th row and j th column of P . Let P_i denote the i th row of P . The eigenvalues of P are ordered with nonincreasing magnitude (i.e., $|\lambda_1(P)| \geq \dots \geq |\lambda_n(P)|$). Given two functions $\varphi_1 : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$ and $\varphi_2 : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$, $\varphi_1(n)$ is $O(\varphi_2(n))$ if there exist positive constants c and N such that $|\varphi_1(n)| \leq c|\varphi_2(n)|$ for all $n \geq N$.

II. PROBLEM FORMULATION

We first introduce the following definitions (e.g., [17]).

Definition 1: Given a ground set V , a set function $f : 2^V \rightarrow \mathbb{R}_{\geq 0}$ is said to be monotone nondecreasing if for all $A \subseteq B \subseteq V$, $f(A) \leq f(B)$.

Definition 2: Given a set V , a set function $f : 2^V \rightarrow \mathbb{R}_{\geq 0}$ is said to be submodular if for all $A \subseteq B \subseteq V$ and for all $v \in V \setminus B$, $f(\{v\} \cup A) - f(A) \geq f(\{v\} \cup B) - f(B)$.

Consider a set of n agents that communicate over a graph to maximize a global objective function. Each agent can be viewed as a node $i \in \mathcal{N} \triangleq \{1, \dots, n\}$. Denote the communication graph as $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, which is assumed to be undirected and connected throughout this paper. An edge $(i, j) \in \mathcal{E}$, which is an unordered pair, indicates a *bidirectional* communication between agents i and j , i.e., agent i can receive information from agent j at each time step, and vice versa. Let \mathcal{N}_i denote the set of neighbors of agent i , i.e., $\mathcal{N}_i \triangleq \{j \in \mathcal{N} : (i, j) \in \mathcal{E}\}$.

Definition 3: The diameter of a connected graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ is given by $d(\mathcal{G}) = \max_{i, j \in \mathcal{N}} l_{ij}$, where l_{ij} is the length of the shortest path (i.e., distance) from i to j in \mathcal{G} .

We now consider the scenario where each agent $i \in \mathcal{N}$ has access to a local set function $f_i : 2^V \rightarrow \mathbb{R}_{\geq 0}$ with a cardinality constraint $K \in \mathbb{Z}_{\geq 1}$. The function $f_i(\cdot)$ is assumed to be monotone nondecreasing and submodular for all $i \in \mathcal{N}$.¹ Meanwhile, we assume without loss of generality that $f_i(\cdot)$ is normalized such that $f_i(\emptyset) = 0$ for all $i \in \mathcal{N}$. The objective for the agents is to solve, in a distributed manner (i.e., by repeatedly exchanging information only with their neighbors), the following global optimization problem:

$$\max_{S \subseteq V, |S| \leq K} f(S) = \max_{S \subseteq V, |S| \leq K} \frac{1}{n} \sum_{i=1}^n f_i(S), \quad (1)$$

where $f(S) \triangleq \frac{1}{n} \sum_{i=1}^n f_i(S)$, and $K \in \mathbb{Z}_{\geq 1}$.

Since the sum of monotone nondecreasing submodular functions is monotone nondecreasing submodular, $f(\cdot)$ is monotone nondecreasing submodular (with $f(\emptyset) = 0$). Thus, the global optimization problem (Problem (1)) is to maximize a monotone nondecreasing submodular function subject to a cardinality constraint. A (centralized) greedy algorithm has been proposed to solve Problem (1) with a multiplicative approximation ratio of $(1 - 1/e)$ [17]. Moreover, it was shown in [18] that the greedy algorithm achieves the best possible approximation ratio of any polynomial-time approximation algorithm for Problem (1) in the centralized case if $P \neq NP$. This motivates us to extend the centralized greedy algorithm to solve Problem (1) in the distributed case.

III. DISTRIBUTED GREEDY ALGORITHM

The main idea of the distributed greedy algorithm (Algorithm 1) is as follows. Based on the centralized greedy algorithm, the distributed greedy algorithm runs for K rounds in total. Given a current set S_k ($|S_k| = k$) of selected elements before the $(k+1)$ th ($k \leq K-1$) round of the algorithm,

¹We will generalize our analysis to monotone nondecreasing nonsubmodular functions later.

the $(k+1)$ th round of the algorithm lets all the agents reach consensus at an element $s' \in V \setminus S_k$ (after a certain number of communication steps) that maximizes $(f(\{s'\} \cup S_k) - f(S_k))$ with some additive error (suboptimality) and add s' to S_k to obtain S_{k+1} . After K rounds, all the agents obtain a set S_K ($|S_K| = K$) that gives a solution to Problem (1) with a suboptimality bound which we will discuss later.

Algorithm 1 Distributed Greedy Algorithm

Input: $f_i : 2^V \rightarrow \mathbb{R}_{\geq 0}$, $K \in \mathbb{Z}_{\geq 1}$, $\mathcal{G} = (\mathcal{N}, \mathcal{E})$
Design parameters: $W \in \mathbb{R}^{n \times n}$, $T \in \mathbb{Z}_{\geq 1}$, $T' = T + 1 + d(\mathcal{G})$, $\psi \in \mathbb{R}_{\geq 0}$
Output: \bar{S}_i^K

```

1:  $k = 0$ ,  $\bar{S}_i^0 = \emptyset$ 
2: while  $k \leq K-1$  do
3:   Calculate  $x_{i,v}^0 = f_i(\{v\} \cup \bar{S}_i^k) - f_i(\bar{S}_i^k)$ ,  $\forall v \in V \setminus \bar{S}_i^k$ 
4:   Stack  $x_{i,v}^0$  into a vector  $x_i^0 \in \mathbb{R}^{|V \setminus \bar{S}_i^k|}$ 
5:   for  $t = 0, \dots, T-1$  do
6:     Update  $x_i^{t+1} = w_{ii}x_i^t + \sum_{j \in \mathcal{N}_i} w_{ij}x_j^t$ 
7:   end for
8:   Find  $x_{i,v_{i^*}}^T = \max_{v \in V} x_{i,v}^T$ 
9:   Set  $S_i^{T+1} = \{v : x_{i,v}^T \geq x_{i,v_{i^*}}^T - \psi\}$ 
10:  for  $t = T+1, \dots, T'-1$  do
11:    Update  $S_i^{t+1} = \bigcap_{j \in \mathcal{N}_i} S_j^t$ 
12:  end for
13:  Find  $j_{\min} = \min\{j : v_j \in S_i^{T'}\}$ 
14:  Update  $\bar{S}_i^{k+1} = \{v_{j_{\min}}\} \cup \bar{S}_i^k$ 
15:   $k = k + 1$ 
16: end while
17:  $\bar{S}_i^K = \bar{S}_i^k$ 

```

Specifically, denote $V \triangleq \{v_1, \dots, v_{|V|}\}$. Each agent $i \in \mathcal{N}$ initializes a local variable $\bar{S}_i^k = \emptyset$ with $k = 0$. In the first round of the distributed greedy algorithm (with $k = 0$), each agent $i \in \mathcal{N}$ starts with $t = 0$, and calculates $x_{i,v}^0 = f_i(\{v\}) - f_i(\emptyset)$ for all $v \in V$, where $f_i(\emptyset) = 0$. In other words, each agent $i \in \mathcal{N}$ maintains a local variable $x_i^t = [x_{i,v_1}^t \ \dots \ x_{i,v_{|V|}}^t]^T \in \mathbb{R}^{|V|}$ at time step $t = 0$. Given design parameter $T \in \mathbb{Z}_{\geq 1}$ of the algorithm, agent i first updates x_i^t (using x_i^t and x_j^t for all $j \in \mathcal{N}_i$) from time step $t = 0$ to time step $t = T$. Then, agent i obtains a set S_i^{T+1} and maintains a local variable S_i^t . Given design parameter $T' \in \mathbb{Z}_{\geq 1}$ of the algorithm, where $T' > T+1$,² agent i now updates S_i^t (using S_j^t for all $j \in \mathcal{N}_i$) from time step $t = T+1$ to time step $t = T'$. At the end of the first round of the distributed greedy algorithm (i.e., at $t = T'$), all the agents in \mathcal{N} choose the same element s_0 from $S_i^{T'}$ and update \bar{S}_i^0 as $\bar{S}_i^1 = \bar{S}_i^0 \cup \{s_0\}$. After finishing the first round, the distributed greedy algorithm enters the second round (with $k = 1$). Again, each agent $i \in \mathcal{N}$ starts with $t = 0$ and calculates $x_i^0 = [x_{i,v_{j_1}}^0 \ \dots \ x_{i,v_{|V_1|}}^0]^T \in \mathbb{R}^{|V_1|}$, where $V_1 \triangleq V \setminus \bar{S}_i^1 = \{v_{j_1}, \dots, v_{j_{|V_1|}}\}$ and $x_{i,v}^0 = f_i(\{v\} \cup \bar{S}_i^1) - f_i(\bar{S}_i^1)$ for all $v \in V_1$. Similarly to the first round of the distributed greedy

²We will explain the choice of T' in the algorithm later.

algorithm, agent i obtains the updated local variables x_i^T and $S_i^{T'}$, which leads to the update of \bar{S}_i^1 as $\bar{S}_i^2 = \bar{S}_i^1 \cup \{s_1\}$, where s_1 is an element chosen from $S_i^{T'}$ by all the agents in \mathcal{N} . The distributed greedy algorithm repeats the above process for K rounds. Note from the above arguments that for all $k \in \{0, \dots, K-1\}$, $\bar{S}_i^k = \bar{S}_i^k$ for all $i, j \in \mathcal{N}$.

Remark 1: As we will see in the following, our distributed greedy algorithm has a consensus phase (with a limited number of communications) among the set of agents in each round of the algorithm. Similar consensus-based distributed algorithms have been used in, for example, distributed task allocation [19] and distributed Kalman filtering [20].

To implement Algorithm 1, we assume the following.

Assumption 1: The set \mathcal{N} of agents has a synchronized clock such that all the agents in \mathcal{N} know the current values of $k \in \mathbb{Z}_{\geq 0}$ and $t \in \mathbb{Z}_{\geq 1}$. Moreover, the agents in \mathcal{N} know the design parameters T and T' before running the algorithm, where $T, T', K \in \mathbb{Z}_{\geq 1}$ and $T' > T + 1$.

We now describe (distributed) update rules for the agents in \mathcal{N} to update the local variables x_i^t , S_i^t and \bar{S}_i^k from Algorithm 1, respectively. Consider the $(k+1)$ th round of the distributed algorithm, where $k \leq K-1$ ($k \in \mathbb{Z}_{\geq 0}$). Noting that $\bar{S}_i^k = \bar{S}_j^k$ for all $i, j \in \mathcal{N}$, we denote $\bar{S}^k = \bar{S}_i^k$.

First, we describe the update rule for $x_i^t \in \mathbb{R}^{|V_k|}$, where $V_k \triangleq V \setminus \bar{S}_i^k$. Note that each agent $i \in \mathcal{N}$ initializes x_i^0 in lines 3-4 of Algorithm 1 as

$$x_i^0 = \begin{bmatrix} x_{i,v_{l_1}}^0 & \dots & x_{i,v_{l_{|V_k|}}}^0 \end{bmatrix}' \in \mathbb{R}^{|V_k|}, \quad (2)$$

where $x_{i,v}^0 = f_i(\{v\} \cup \bar{S}_i^k) - f_i(\bar{S}_i^k)$ for all $v \in V_k$ and $V_k = \{v_{l_1}, \dots, v_{l_{|V_k|}}\}$. Since $f_i(\cdot)$ is monotone nondecreasing for all $i \in \mathcal{N}$, we have $x_{i,v}^0 \geq 0$ for all $i \in \mathcal{N}$ and for all $v \in V_k$. Starting from time step $t = 0$ with x_i^0 , each agent $i \in \mathcal{N}$ computes x_i^{t+1} in line 6 of Algorithm 1 according to the following update rule:

$$x_i^{t+1} = w_{ii}x_i^t + \sum_{j \in \mathcal{N}_i} w_{ij}x_j^t, \quad (3)$$

where agent i assigns a weight $w_{ij} \in \mathbb{R}$ to agent j for all $j \in \mathcal{N}_i \cup \{i\}$ and updates x_i^{t+1} as a weighted average of x_j^t from $j \in \mathcal{N}_i \cup \{i\}$. Denote $W \in \mathbb{R}^{n \times n}$ as the weight matrix (or *mixing matrix*) such that $W_{ij} = w_{ij}$ for all $i, j \in \mathcal{N}$. We assume that the weight matrix W satisfies the following assumptions, which are standard in the distributed optimization literature (e.g., [5]).

Assumption 2: The weight matrix $W \in \mathbb{R}^{n \times n}$ (associated with $\mathcal{G} = (\mathcal{N}, \mathcal{E})$) is assumed to satisfy: (1) $W_{ij} \in \mathbb{R}_{\geq 0}$ for all $i, j \in \mathcal{N}$ and $W_{ij} = 0$ if $(i, j) \notin \mathcal{E}$; (2) $W\mathbf{1}_n = \mathbf{1}_n$; (3) $W = W'$ and (4) $\mu(W) \triangleq \max\{\lambda_2(W), -\lambda_n(W)\} < 1$.

Remark 2: Assumption 2.(1)-(3) ensure that W is symmetric and doubly stochastic. Thus, the eigenvalues of W are real and satisfy $1 = \lambda_1(W) \geq \lambda_2(W) \geq \dots \geq \lambda_n(W) \geq -1$ (e.g., [21]). Assumption 2.(4) is satisfied if the Markov chain corresponding to matrix W is irreducible and aperiodic (e.g., [21], [22]). Note that the weight matrix W is also a design parameter of the distributed greedy algorithm.

Similarly to Assumption 1, we assume that each agent $i \in \mathcal{N}$ knows W_i (i.e., w_{ij} for all $j \in \mathcal{N} \cup \{i\}$).

By repeatedly running update rule (3), x_i^t will converge to $\frac{1}{n} \sum_{j \in \mathcal{N}} x_j^0$ as $t \rightarrow \infty$ (e.g., [21]) for all $i \in \mathcal{N}$, so that $x_{i,v}^t$ will converge to $\frac{1}{n} \sum_{j \in \mathcal{N}} x_{j,v}^0$ as $t \rightarrow \infty$ for all $v \in V_k$. To analyze finite-time performance of update rule (3), let us first consider the following function of $t \in \mathbb{Z}_{\geq 1}$:

$$\delta_k(t) = \max_{i \in \mathcal{N}, v \in V_k} \left| x_{i,v}^t - \frac{1}{n} \sum_{j \in \mathcal{N}} x_{j,v}^0 \right|, \quad (4)$$

which is the maximum deviation of $x_{i,v}^t$ from the limiting value $\frac{1}{n} \sum_{j \in \mathcal{N}} x_{j,v}^0$ over all agents $i \in \mathcal{N}$ and all elements $v \in V_k$ at any (finite) time step t . Moreover, note that for each agent $i \in \mathcal{N}$ and an element $v \in V_k$, we can view $x_{i,v}^t$ as an estimate of $\frac{1}{n} \sum_{j \in \mathcal{N}} (f_j(\{v\} \cup \bar{S}_i^k) - f_j(\bar{S}_i^k))$ at time step t . Thus, $\delta_k(t)$ captures the maximum error (in absolute value) of such estimates over all $i \in \mathcal{N}$ and all $v \in V_k$ at time step t . We will use the following result (e.g., [22]).

Lemma 1: Consider a weight matrix $W \in \mathbb{R}^{n \times n}$ that satisfies Assumption 2. The following inequality holds:

$$\max_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}} \left| (W^t)_{ij} - \frac{1}{n} \right| \leq \sqrt{n}(\mu(W))^t,$$

where $\mu(W) = \max\{\lambda_2(W), -\lambda_n(W)\}$. \square

We then have the following results.

Lemma 2: Consider the update rule (3) initialized with x_i^0 given by Eq. (2). Suppose Assumptions 1 and 2 hold. For all time steps $t \in \mathbb{Z}_{\geq 1}$, the following inequality holds:

$$\delta_k(t) \leq \sqrt{n}(\mu(W))^t F_h, \quad (5)$$

where $\delta_k(t)$ is defined in Eq. (4) and $F_h \triangleq \max_{i \in \mathcal{N}} f_i(V)$. \square

Proof: Denoting $\bar{x}_v^t = [x_{1,v}^t \dots x_{n,v}^t]'$ for all $v \in V_k$ and for all $t \in \mathbb{Z}_{\geq 0}$, we have from Eq. (3) $\bar{x}_v^{t+1} = W\bar{x}_v^t$, which implies $\bar{x}_v^t = W^t\bar{x}_v^0$. It then follows that

$$\begin{aligned} \delta_k(t) &= \max_{i \in \mathcal{N}, v \in V_k} \left| (W^t)_{iv} \bar{x}_v^0 - \frac{1}{n} \sum_{j \in \mathcal{N}} x_{j,v}^0 \right| \\ &= \max_{i \in \mathcal{N}, v \in V_k} \left| \sum_{j \in \mathcal{N}} (W^t)_{ij} x_{j,v}^0 - \frac{1}{n} \sum_{j \in \mathcal{N}} x_{j,v}^0 \right| \\ &= \max_{v \in V_k} \max_{i \in \mathcal{N}} \left| \sum_{j \in \mathcal{N}} ((W^t)_{ij} - \frac{1}{n}) x_{j,v}^0 \right| \\ &\leq \max_{v \in V_k} \max_{i \in \mathcal{N}} \left(\sum_{j \in \mathcal{N}} \left| (W^t)_{ij} - \frac{1}{n} \right| x_{j,v}^0 \right) \\ &\leq \sqrt{n}(\mu(W))^t \max_{j \in \mathcal{N}} \max_{v \in V_k} x_{j,v}^0 \leq \sqrt{n}(\mu(W))^t F_h, \end{aligned} \quad (6)$$

where the first inequality in (6) follows from Lemma 1. For the second inequality in (6), we note that $x_{j,v}^0 = f_j(\{v\} \cup \bar{S}_i^k) - f_j(\bar{S}_i^k) \leq f_j(V)$ for all $j \in \mathcal{N}$ and for all $v \in V_k$, since $f_j(\cdot)$ is monotone nondecreasing with $f_j(\emptyset) = 0$.³ Thus, we have $\max_{j \in \mathcal{N}} \max_{v \in V_k} x_{j,v}^0 \leq \max_{j \in \mathcal{N}} f_j(V) = F_h$. \blacksquare

³Noting that $f_j(\{v\} \cup \bar{S}_i^k) - f_j(\bar{S}_i^k) \leq f_j(v) - f_j(\emptyset) \leq f_j(V)$ $\forall j \in \mathcal{N}$ and $\forall v \in V_k$ by the submodularity of $f_j(\cdot)$, the bound in (5) can potentially be tightened by defining $F'_h \triangleq \max_{i \in \mathcal{N}} \max_{v \in V} f_i(v)$.

Lemma 3: Consider the update rule (3) initialized with x_i^0 given by Eq. (2). Suppose Assumptions 1 and 2 hold. For each time step $t \in \mathbb{Z}_{\geq 1}$, denote $x_{i,v_{i^*}}^t = \max_{v \in V_k} x_{i,v}^t$, where $v_i^* \triangleq \arg \max_{v \in V_k} x_{i,v}^t$, for all $i \in \mathcal{N}$. The following holds:

$$x_{i,v_{j^*}}^t \geq x_{i,v_{i^*}}^t - 4\epsilon(t), \forall i, j \in \mathcal{N}, \quad (7)$$

where

$$\epsilon(t) \triangleq \sqrt{n}(\mu(W))^t F_h \quad (8)$$

is a function of $t \in \mathbb{Z}_{\geq 1}$, and $F_h = \max_{i \in \mathcal{N}} f_i(V)$. \square

Proof: Consider any time step $t \in \mathbb{Z}_{\geq 1}$, and any two (distinct) agents $i, j \in \mathcal{N}$. We have from Lemma 2

$$\left| x_{i,v_{i^*}}^t - \frac{1}{n} \sum_{q \in \mathcal{N}} x_{q,v_{i^*}}^0 \right| \leq \epsilon(t), \quad (9)$$

and

$$\left| x_{j,v_{i^*}}^t - \frac{1}{n} \sum_{q \in \mathcal{N}} x_{q,v_{i^*}}^0 \right| \leq \epsilon(t). \quad (10)$$

It then follows from (9)-(10) that

$$\left| x_{i,v_{i^*}}^t - x_{j,v_{i^*}}^t \right| \leq 2\epsilon(t). \quad (11)$$

Similarly, we have

$$\left| x_{j,v_{j^*}}^t - x_{i,v_{j^*}}^t \right| \leq 2\epsilon(t). \quad (12)$$

Therefore, we have the following:

$$x_{i,v_{i^*}}^t - x_{i,v_{j^*}}^t \leq x_{j,v_{i^*}}^t + 2\epsilon(t) - x_{j,v_{j^*}}^t + 2\epsilon(t) \leq 4\epsilon(t),$$

where the first inequality follows from (11)-(12) and the second inequality follows from the fact $x_{j,v_{j^*}}^t = \max_{v \in V_k} x_{j,v}^t$, i.e. $x_{j,v_{i^*}}^t \leq x_{j,v_{j^*}}^t$. \blacksquare

Note that each agent $i \in \mathcal{N}$ updates x_i^t from time step $t = 0$ to time step $t = T$, where we recall from Assumption 1 that $T \in \mathbb{Z}_{\geq 1}$ is a design parameter of the algorithm that is known to all the agents in \mathcal{N} .

Next, we describe the update rule for S_i^t . Specifically, after running update rule (3) from time step $t = 0$ to time step $t = T$ and obtaining x_i^T , each agent $i \in \mathcal{N}$ obtains S_i^{T+1} as

$$S_i^{T+1} = \{v : x_{i,v}^T \geq x_{i,v_{i^*}}^T - \psi\}, \quad (13)$$

where $\psi \in \mathbb{R}_{\geq 0}$ is a design parameter of the algorithm that needs to satisfy the following condition:

$$\psi \geq 4\sqrt{n}(\mu(W))^T F_h. \quad (14)$$

Note that we also assume that each agent $i \in \mathcal{N}$ knows the design parameter ψ . We then see from Lemma 3 and update rule (13) with condition (14) that $v_{j^*} \in S_i^{T+1}$ for all $j \in \mathcal{N}$, where $v_{j^*} = \arg \max_{v \in V_k} x_{j,v}^T$. This implies that $V^* \subseteq S_i^{T+1}$ for all $i \in \mathcal{N}$, where $V^* \triangleq \{v_{1^*}, \dots, v_{n^*}\}$. Starting from S_i^{T+1} at time step $t = T + 1$, each agent $i \in \mathcal{N}$ computes S_i^t according to the following update rule:

$$S_i^{t+1} = \bigcap_{j \in \mathcal{N}_i} S_j^t. \quad (15)$$

We will use the following result whose proof follows directly from induction and is thus omitted for conciseness.

Lemma 4: Consider the communication graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ and the update rule (15) initialized with S_i^{T+1} given by Eq. (13). Suppose Assumption 1 holds. For each $t \geq T + 1 + d(\mathcal{G})$ ($t \in \mathbb{Z}$), $S_i^t = \bigcap_{j \in \mathcal{N}} S_j^{T+1}$ for all $i \in \mathcal{N}$, where $d(\mathcal{G})$ is given by Definition 3. \square

Given any $T \in \mathbb{Z}_{\geq 1}$, we then set the design parameter $T' = T + 1 + d(\mathcal{G})$ in the sequel. After running update rule (15) until time step $t = T'$ (starting from time step $t = T + 1$), we have from Lemma 4 $S_i^{T'} = S_j^{T'}$ for all $i, j \in \mathcal{N}$. Moreover, noting from the above arguments that $V^* \subseteq S_i^{T+1}$ for all $i \in \mathcal{N}$, where $V^* = \{v_{1^*}, \dots, v_{n^*}\}$, we have $V^* \subseteq S_i^{T'}$, i.e., $S_i^{T'} \neq \emptyset$. Finally, denoting $j_{\min} = \min\{j : v_j \in S_i^{T'}\}$, each agent $i \in \mathcal{N}$ updates \bar{S}_i^k as

$$\bar{S}_i^{k+1} = \{v_{j_{\min}}\} \cup \bar{S}_i^k. \quad (16)$$

Noting that $\bar{S}_i^k = \bar{S}_j^k$ for all $i, j \in \mathcal{N}$, we obtain $\bar{S}_i^{k+1} = \bar{S}_j^{k+1}$ for all $i, j \in \mathcal{N}$. Denote $\bar{S}^{k+1} = \bar{S}_i^{k+1}$. Combining update rules (3) and (13)-(16) leads to the following result.

Lemma 5: Consider the update rules (3), (13) and (15)-(16), where (3) is initialized with x_i^0 given by Eq. (2). Suppose Assumptions 1 and 2 hold and $\psi \in \mathbb{R}_{\geq 0}$ satisfies condition (14). Then

$$f(\bar{S}^{k+1}) - f(\bar{S}^k) \geq \left\{ \max_{v \in V_k} f(\{v\} \cup \bar{S}^k) - f(\bar{S}^k) \right\} - \psi - 2\epsilon(T), \quad (17)$$

where $\epsilon(T) = \sqrt{n}(\mu(W))^T F_h$, and $F_h = \max_{i \in \mathcal{N}} f_i(V)$. \square

Proof: Denote $v^* \triangleq \arg \max_{v \in V_k} f(\{v\} \cup \bar{S}^k) - f(\bar{S}^k)$ and note that $\bar{S}^{k+1} \setminus \bar{S}^k = v_{j_{\min}}$, where $j_{\min} = \min\{j : v_j \in S_i^{T'}\}$. We see from Lemma 2 and the definition of $x_{j,v}^0$ that

$$\left| x_{i,v^*}^T - (f(\{v^*\} \cup \bar{S}^k) - f(\bar{S}^k)) \right| \leq \epsilon(T), \quad (18)$$

and

$$\left| x_{i,v_{j_{\min}}}^T - (f(\{v_{j_{\min}}\} \cup \bar{S}^k) - f(\bar{S}^k)) \right| \leq \epsilon(T). \quad (19)$$

Noting that $S_i^{T'} \subseteq S_i^{T+1}$ and $v_{j_{\min}} \in S_i^{T'}$, we have from update rule (13)

$$x_{i,v_{j_{\min}}}^T \geq x_{i,v_{i^*}}^T - \psi, \quad (20)$$

where $x_{i,v_{i^*}}^T = \max_{v \in V_k} x_{i,v}^T$. We then have the following:

$$(f(\{v_{j_{\min}}\} \cup \bar{S}^K) - f(\bar{S}^k)) - (f(\{v^*\} \cup \bar{S}^k) - f(\bar{S}^k))$$

$$\geq (f(\{v_{j_{\min}}\} \cup \bar{S}^K) - f(\bar{S}^k)) - x_{i,v^*}^T - \epsilon(T) \quad (21)$$

$$\geq (f(\{v_{j_{\min}}\} \cup \bar{S}^K) - f(\bar{S}^k)) - x_{i,v_{i^*}}^T - \epsilon(T) \quad (22)$$

$$\geq (f(\{v_{j_{\min}}\} \cup \bar{S}^K) - f(\bar{S}^k)) - x_{i,v_{j_{\min}}}^T - \psi - \epsilon(T) \quad (23)$$

$$\geq -\psi - 2\epsilon(T), \quad (24)$$

where (21) and (24) follow from (18) and (19), respectively, (23) follows from (20), and (22) follows from the fact $x_{i,v_{i^*}}^T = \max_{v \in V_k} x_{i,v}^T$. \blacksquare

In summary, after running (3), (13) and (15)-(16) as described above in the $(k+1)$ th round of the distributed greedy algorithm, each agent $i \in \mathcal{N}$ obtains x_i^T , $S_i^{T'}$ and

⁴Note that all the agents in \mathcal{N} label the elements in V such that v_j refers to the same element in V for all $j \in \{1, \dots, m\}$.

\bar{S}_i^{k+1} , where $T' = T + 1 + d(\mathcal{G})$ and $\bar{S}_i^{k+1} = \bar{S}_j^{k+1}$ for all $i, j \in \mathcal{N}$. The algorithm then enters the next round (with k incremented by 1) and repeats the same processes as described above, where all the results derived still hold.

The procedure described in this section is summarized in Algorithm 1, where the algorithm is implemented at each $i \in \mathcal{N}$ in a distributed way. As argued above, Algorithm 1 allows all the agents in $i \in \mathcal{N}$ to reach consensus at a solution \bar{S}^K to Problem (1), i.e., $\bar{S}_i^K = \bar{S}^K \forall i \in \mathcal{N}$.

IV. CONVERGENCE ANALYSIS

In this section, we analyze the performance (i.e., convergence) of Algorithm 1. Note that the (centralized) greedy algorithm solves Problem (1) in the centralized case with the multiplicative approximation ratio of $(1 - 1/e)$, i.e., $f(S_g) \geq (1 - 1/e)f(S^*)$, where S_g is the solution returned by the greedy algorithm and S^* is an optimal solution to Problem (1). Hence, we analyze the performance of Algorithm 1 by comparing the convergence of $f(\bar{S}_i^K)$ to $(1 - 1/e)f(S^*)$. We will use the following result from [23].

Lemma 6: Consider Problem (1) in the centralized case. Denote $\bar{G}_1 \triangleq \emptyset$ and $\bar{G}_j \triangleq \{\bar{g}_1, \dots, \bar{g}_{j-1}\}$ for all $j \in \{2, \dots, K+1\}$. Suppose

$$f(\{\bar{g}_j\} \cup \bar{G}_j) - f(\bar{G}_j) \geq \left\{ \max_{v \in V \setminus \bar{G}_j} (f(\{v\} \cup \bar{G}_j) - f(\bar{G}_j)) \right\} - \tau_j$$

for all $j \in \{1, \dots, K\}$, where $\tau_j \in \mathbb{R}_{\geq 0}$. Then

$$f(\bar{G}_{K+1}) \geq (1 - \frac{1}{e})f(S^*) - \sum_{k=1}^K \tau_k, \quad (25)$$

where S^* is an optimal solution to Problem (1). \square

The result below follows directly from Lemmas 5 and 6.

Theorem 1: Consider Algorithm 1 for Problem (1) with a set \mathcal{N} of agents. Suppose Assumptions 1 and 2 hold and $\psi \in \mathbb{R}_{\geq 0}$ satisfies condition (14). Then Algorithm 1 lets all the agents in \mathcal{N} reach consensus at a solution \bar{S}^K to Problem (1) that satisfies

$$f(\bar{S}^K) \geq (1 - \frac{1}{e})f(S^*) - K(\psi + 2\epsilon(T)),$$

where $\epsilon(T) = \sqrt{n}(\mu(W))^T F_h$ with $F_h = \max_{i \in \mathcal{N}} f_i(V)$, and S^* is an optimal solution to Problem (1). \square

Theorem 1 shows that \bar{S}^K approaches the $(1 - 1/e)$ approximation of S^* with an additive factor $E_r \triangleq K(\psi + 2\epsilon(T))$. We know from the definition of $\epsilon(T)$ that E_r is a function of the number of the agents, the bound on the local functions and the design parameters (i.e., W , T and ψ). In the context of Theorem 1, we analyze how the additive factor (i.e., E_r) behaves in terms of those quantities under different scenarios. In particular, we are interested in how the additive factor depends on the number of communication steps in each round of the algorithm. First, let us consider the fast communication scenario (e.g., [20]). In this scenario, agents can communicate sufficiently fast, i.e., $T \rightarrow \infty$, in each round of the distributed greedy algorithm. Since $\mu(W) < 1$ from Assumption 2, $\epsilon(T) \rightarrow 0$ as $T \rightarrow \infty$. Moreover, the lower bound on ψ in (14) tends to zero. Consequently, we

can choose the design parameter ψ to be arbitrarily close to zero and obtain $E_r \rightarrow 0$.

Next, we consider the scenario where the communication among the agents in each round of the distributed algorithm is limited. Suppose n is fixed and the input to Algorithm 1 is also fixed, i.e., K and F_h are fixed. We then have $E_r = K\psi + O((\mu(W))^T)$. If we can choose the design parameter ψ such that $\psi = O((\mu(W))^T)$, we obtain $E_r = O((\mu(W))^T)$, which implies that E_r vanishes at an exponential rate. In contrast, if we assume that ψ is fixed, we have $E_r = K\psi + O((\mu(W))^T)$, which implies that E_r converges exponentially to $K\psi$.

Indeed, using techniques in, e.g., [21], one can optimally choose the weight matrix W such that $\mu(W)$ is minimized in the above scenarios, which leads to accelerations in the convergence rate. In summary, we observe a tradeoff between the performance of the distributed greedy algorithm and the number of communication steps in each round of the algorithm. Moreover, the performance of the algorithm also depends on the choice of ψ . It is also worth noting that our distributed greedy algorithm achieves exponential convergence rates (as described above), while the algorithm proposed in [6] only achieves sublinear convergence rates.

V. NONSUBMODULAR OBJECTIVE FUNCTIONS

In this section, we extend our previous analysis to cases when the objective functions in Problem (1) are nonsubmodular. In other words, we consider the scenario where the local objective function $f_i(\cdot)$ is monotone nondecreasing with $f_i(\emptyset) = 0$, but not necessarily submodular, for all $i \in \mathcal{N}$. We first note that the (centralized) greedy algorithm has also been applied to solve Problem (1) with nonsubmodular objective functions using the notion of submodularity ratio (e.g., [15]).

Definition 4: (Submodularity ratio) Given a set V , the submodularity ratio of a nonnegative set function $f : 2^V \rightarrow \mathbb{R}_{\geq 0}$ is the largest $\gamma \in \mathbb{R}_{\geq 0}$ that satisfies $\sum_{a \in A \setminus B} (f(\{a\} \cup B) - f(B)) \geq \gamma(f(A \cup B) - f(B))$ for all $A, B \subseteq V$.

Remark 3: For a nonnegative and nondecreasing function set $f(\cdot)$ with submodularity ratio γ , we have $\gamma \in [0, 1]$, and $f(\cdot)$ is submodular if and only if $\gamma = 1$ [15].

We now extend Lemma 6 to nonsubmodular functions; a proof of the following result is included in the appendix.

Lemma 7: Consider Problem (1) in the centralized case, where the objective function $f(\cdot)$ is monotone nondecreasing with submodularity ratio $\gamma \in \mathbb{R}_{>0}$. Denote $\bar{G}_1 \triangleq \emptyset$ and $\bar{G}_j \triangleq \{\bar{g}_1, \dots, \bar{g}_{j-1}\}$ for all $j \in \{2, \dots, K+1\}$. Suppose

$$f(\{\bar{g}_j\} \cup \bar{G}_j) - f(\bar{G}_j) \geq \left\{ \max_{v \in V \setminus \bar{G}_j} (f(\{v\} \cup \bar{G}_j) - f(\bar{G}_j)) \right\} - \tau_j$$

for all $j \in \{1, \dots, K\}$, where $\tau_j \in \mathbb{R}_{\geq 0}$. Then

$$f(\bar{G}_{K+1}) \geq (1 - e^{-\gamma})f(S^*) - \sum_{k=1}^K \tau_k, \quad (26)$$

where S^* is an optimal solution to Problem (1). \square

Using similar arguments to those for Theorem 1, one can obtain the following result from Definition 4 and Lemma 7; the proof is omitted for conciseness.

Corollary 1: Consider Algorithm 1 for Problem (1) with a set \mathcal{N} of agents. Suppose Assumptions 1 and 2 hold and $\psi \in \mathbb{R}_{\geq 0}$ satisfies condition (14). Denote the submodularity ratio of the local objective function $f_i(\cdot)$ as $\gamma_i \in \mathbb{R}$ for all $i \in \mathcal{N}$ and denote $\gamma_c \triangleq \min_{i \in \mathcal{N}} \gamma_i$. Suppose $\gamma_i > 0$ for all $i \in \mathcal{N}$. Then Algorithm 1 lets all the agents in \mathcal{N} reach consensus at a solution \bar{S}^K to Problem (1) that satisfies

$$f(\bar{S}^K) \geq (1 - e^{-\gamma_c})f(S^*) - K(\psi + 2\epsilon(T)),$$

where $\epsilon(T) = \sqrt{n}(\mu(W))^T F_h$ with $F_h = \max_{i \in \mathcal{N}} f_i(V)$, and S^* is an optimal solution to Problem (1). \square

Similarly to Section IV, Corollary 1 leads to a tradeoff between the performance of the distributed greedy algorithm and the number of communication steps in each round of the algorithm, under the nonsubmodular setting.

VI. CONCLUSIONS

In this paper, we proposed a distributed greedy algorithm for maximizing a global submodular function, subject to a cardinality constraint, with a group of agents communicating over a network. The distributed greedy algorithm allows each agent to converge to within an additive factor of the $(1 - 1/e)$ approximation of the optimal solution to the global maximization problem. The additive factor reveals a tradeoff between the performance of the algorithm and the number of communication steps in each round of the algorithm. Finally, we extended our analysis to cases when the objective function is not submodular by leveraging the notion of submodularity ratio.

APPENDIX

Proof of Lemma 7:

The proof is based on the idea of the proof for Theorem 6 in [23]. Denote $\Delta_j \triangleq f(S^*) - f(\bar{G}_j)$ for all $j \in \{1, \dots, K+1\}$ and $\bar{\beta}_j \triangleq f(\{\bar{g}_j \cup \bar{G}_j\}) - f(\bar{G}_j) = f(\bar{G}_{j+1}) - f(\bar{G}_j)$ for all $j \in \{1, \dots, K\}$. We then have from Definition 4

$$f(\bar{G}_j \cup S^*) - f(\bar{G}_j) \leq \frac{1}{\gamma} \left(\sum_{v \in S^* \setminus \bar{G}_j} f(\{v\} \cup \bar{G}_j) - f(\bar{G}_j) \right). \quad (27)$$

Noting that $\bar{\beta}_j \geq \{\max_{v \in V \setminus \bar{G}_j} (f(\{v\} \cup \bar{G}_j) - f(\bar{G}_j))\} - \tau_j$ for all $j \in \{1, \dots, K\}$, we have $f(\{v\} \cup \bar{G}_j) - f(\bar{G}_j) \leq \bar{\beta}_j + \tau_j$ for all $v \in S^* \setminus \bar{G}_j$ and for all $j \in \{1, \dots, K\}$. It then follows from $|S^*| = K$ and (27) that

$$\begin{aligned} f(S^*) &\leq f(\bar{G}_j \cup S^*) \leq f(\bar{G}_j) + \frac{K}{\gamma}(\bar{\beta}_j + \tau_j) \\ &\Rightarrow \Delta_j \leq \frac{K}{\gamma}(\bar{\beta}_j + \tau_j) \Rightarrow \Delta_j \leq \frac{K}{\gamma}(\Delta_j - \Delta_{j+1} + \tau_j) \\ &\Rightarrow \Delta_{j+1} \leq (1 - \frac{\gamma}{K})\Delta_j + \tau_j. \end{aligned} \quad (28)$$

Unrolling (28), we obtain $\Delta_{K+1} \leq (1 - \frac{\gamma}{K})^K \Delta_1 + \sum_{j=1}^K \tau_j$, where we use the fact $1 - \frac{\gamma}{K} < 1$. Therefore, $f(S^*) - f(\bar{G}_{K+1}) \leq (1 - \frac{\gamma}{K})^K f(S^*) + \sum_{j=1}^K \tau_j$, which implies $f(\bar{G}_{K+1}) \geq f(S^*) - e^{-\gamma} f(S^*) - \sum_{j=1}^K \tau_j$. \blacksquare

REFERENCES

- [1] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Trans. on Automatic Control*, vol. 31, no. 9, pp. 803–812, 1986.
- [2] L. Xiao and S. Boyd, "Optimal scaling of a gradient method for distributed resource allocation," *Journal of Optimization Theory and Applications*, vol. 129, no. 3, pp. 469–488, 2006.
- [3] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Trans. on Automatic Control*, vol. 54, no. 1, p. 48, 2009.
- [4] A. Nedic, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Trans. on Automatic Control*, vol. 55, no. 4, pp. 922–938, 2010.
- [5] K. Yuan, Q. Ling, and W. Yin, "On the convergence of decentralized gradient descent," *SIAM Journal on Optimization*, vol. 26, no. 3, pp. 1835–1854, 2016.
- [6] A. Mokhtari, H. Hassani, and A. Karbasi, "Decentralized submodular maximization: Bridging discrete and continuous settings," in *Proc. of International Conference on Machine Learning*, 2018, pp. 3613–3622.
- [7] B. Gharesifard and S. L. Smith, "Distributed submodular maximization with limited information," *IEEE Trans. on Control of Network Systems*, vol. 5, no. 4, pp. 1635–1645, 2017.
- [8] B. Mirzasoleiman, A. Karbasi, R. Sarkar, and A. Krause, "Distributed submodular maximization: Identifying representative elements in massive data," in *Advances in Neural Information Processing Systems*, 2013, pp. 2049–2057.
- [9] T. Soma, N. Kakimura, K. Inaba, and K.-i. Kawarabayashi, "Optimal budget allocation: Theoretical guarantee and efficient algorithm," in *Proc. of International Conference on Machine Learning*, 2014, pp. 351–359.
- [10] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies," *Journal of Machine Learning Research*, vol. 9, no. Feb, pp. 235–284, 2008.
- [11] S. T. Jawaaid and S. L. Smith, "Submodularity and greedy algorithms in sensor scheduling for linear dynamical systems," *Automatica*, vol. 61, pp. 282–288, 2015.
- [12] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proc. of international conference on Knowledge Discovery and Data mining*. ACM, 2003, pp. 137–146.
- [13] M. L. Fisher, G. L. Nemhauser, and L. A. Wolsey, "An analysis of approximations for maximizing submodular set functions—ii," in *Polyhedral combinatorics*. Springer, 1978, pp. 73–87.
- [14] A. Das and D. Kempe, "Approximate submodularity and its applications: subset selection, sparse approximation and dictionary selection," *Journal of Machine Learning Research*, vol. 19, no. 1, pp. 74–107, 2018.
- [15] A. A. Bian, J. M. Buhmann, A. Krause, and S. Tschiatschek, "Guarantees for greedy maximization of non-submodular functions with applications," in *Proc. of International Conference on Machine Learning*, 2017, pp. 498–507.
- [16] L. Ye and S. Sundaram, "Sensor selection for hypothesis testing: Complexity and greedy algorithms," in *Proc. of Conference on Decision and Control*. IEEE, 2019, pp. 7844–7849.
- [17] 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.
- [18] U. Feige, "A threshold of $\ln n$ for approximating set cover," *Journal of the ACM*, vol. 45, no. 4, pp. 634–652, 1998.
- [19] H.-L. Choi, L. Brunet, and J. P. How, "Consensus-based decentralized auctions for robust task allocation," *IEEE Trans. on Robotics*, vol. 25, no. 4, pp. 912–926, 2009.
- [20] R. Carli, A. Chiuso, L. Schenato, and S. Zampieri, "Distributed Kalman filtering based on consensus strategies," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 4, pp. 622–633, 2008.
- [21] S. Boyd, P. Diaconis, and L. Xiao, "Fastest mixing Markov chain on a graph," *SIAM review*, vol. 46, no. 4, pp. 667–689, 2004.
- [22] P. Diaconis, D. Stroock *et al.*, "Geometric bounds for eigenvalues of Markov chains," *The Annals of Applied Probability*, vol. 1, no. 1, pp. 36–61, 1991.
- [23] M. Streeter and D. Golovin, "An online algorithm for maximizing submodular functions," in *Advances in Neural Information Processing Systems*, 2009, pp. 1577–1584.