On Distributed Submodular Maximization with Limited Information

Bahman Gharesifard Stephen L. Smith

***Abstract*— This paper considers a class of distributed submodular maximization problems in which each agent must choose a single strategy from its strategy set. The objective is to maximize a submodular function of the strategies chosen by each agent. However, each agent has only partial information on the choices of other agents when making its decision. The main objective of this paper is to investigate how the limitation of information about the strategy sets or actions of other agents affects the performance when agents make choices according to a local greedy algorithm. In particular, we provide lower bounds on the performance of greedy algorithms for submodular maximization, which depend on the clique number of the graph. We also characterize graph-theoretic upper bounds in terms of the chromatic number of the graph. Finally, we demonstrate how certain graph properties limit the performance of the greedy algorithm.**

I. Introduction

Many scenarios in multi-agent systems involve situations where the actions of the individual agents contribute to a common objective function, for which each agent has only local information. A well-studied class of such problems is the distributed optimization problem [1], [2], [3], [4], [5], [6], [7], [8], [9], where the main objective is to optimize a sum of some functions, each one available to an individual agent. Unlike this setting, here we consider a collaborative scenario where the agents choose an action from a private discrete strategy set and the goal is to maximize a common objective function defined over the set of actions chosen by each agent. In other words, the total strategy set is partitioned into (disjoint) subsets, each available only to an individual agent. This paper considers a subclass of interest, where the common objective function is additionally assume to be submodular. This assumption holds in a variety of applications of interest, including problems in distributed sensor coverage [10], [11], information gathering [12], and facility location [13].

Building on the early classic work on sumbodular optimization [14], [15], [16], [17], there has been a recent surge of activity on this subject, mainly due to its wide set of applications to areas of computer science, see for example [13] and references therein. It is well known that the submodular maximization problem is NP-hard [16] (unlike submodular minimization, which can be solved in polynomial time [16]). However, good approximation algorithms exist; in particular, when the submodular function is normalized and monotone, a simple greedy algorithm, yields a solution that is within a multiplicative factor of $(1 - 1/\mathrm{e})$ of the optimal. The distributed problem we consider in this paper can be captured as a partition matroid constraint; in particular, there is a bound on the cardinality of the intersection of the solution set and the strategy set of each individual agent. It has been shown [14] that maximizing a normalized and monotone submodular function subject to a matroid constraint can be approximated to within $1/2$ using the simple greedy algorithm. More recently [18] a randomized algorithm has been proposed for this problem, which yields an improved approximation of $(1 - 1/\mathrm{e})$.

The key feature that differentiates the problem under study in this paper from the classic setting of submodular maximization is the limitations on the information available to a decision maker about the actions of others. Similar to the literature on distributed optimization, in order to achieve any reasonable performance, agents need to share information about their actions. The main objective of this paper is to investigate how much information is required to guarantee a certain performance, and what limitations are put on the performance by the topology of the communication graph between agents.

Distributed submodular maximization has recently generated a lot of interest in the context of large-scale maximization [19], [20], [21], where the main goal is to partition the data set into subsets, each of which is maximized separately, and then the overall solution is refined through additional computations by a central node. In our work, however, we envision scenarios where the limitations imposed on individual agents is physical (for example, being able to only estimate data in a neighbourhood). This paper is also somewhat related to the recent work [10], where the role of limitation of information in submodular optimization is studied in the context of coverage problem, for the cases where agents have full information or no information. Here, we address the limitations imposed by the information network topology. Finally, part of our work is related to the so-called "local greedy algorithms", which is studied in the classical paper [15], as we describe in details later. The proofs are omitted due to space limitations and will appear elsewhere.

Statement of Contributions

We consider a class of distributed submodular maximization problems in which each agent has a strategy set from which it must choose a single strategy. The objective is to maximize a submodular function of the set of strategies chosen by each agent. The group of agents take decisions sequentially, having available to them only partial information about the actions takes by the previous agents. The partial na-

Bahman Gharesifard is with the Department of Mathematics and Statistics at Queen's University, Kingston, ON, Canada ghareb@queensu.ca. Stephen L. Smith is with the Department of Electrical and Computer Engineering at the University of Waterloo, Waterloo, ON, Canada stephen.smith@uwaterloo.ca. The work of the authors was partially supported by the Natural Sciences and Engineering Research Council of Canada.

ture of the available information is cast as a directed acyclic graph. Our main objective in this paper is to characterize the fundamental limitations that partial information imposes on the performance of local greedy algorithms. We first show that the well-known $\frac{1}{2}$ lower bound on the performance of greedy algorithms for submodular maximization can be obtained using the so-called clique graph structures, in spite of the fact that none of the agents have access to the whole strategy set. We provide a general lower bound based on the clique number of the graph, and also provide lower bounds for graph topologies with multiple interconnected cliques. We then move on to our next objective, which is characterizing graph-theoretic upper bounds on the performance of the greedy algorithms. We tackle this objective by considering two problem statements. In the first one, we characterize fundamental limitations on the performance of the greedy algorithms on a given graph topology in terms of its chromatic number. Our second result demonstrates how achieving a certain guaranteed performance imposes limitations on the topology of the underlying graph. We characterize the gap between these two bounds, and show how they can be used to efficiently compute limitations of greedy algorithms for scenarios with partial information.

II. Preliminaries

Many combinatorial optimization problems can be phrased as a *submodular maximization problem*. The problem can be stated as follows. Consider a base set of elements E, and let 2^E be the power set of E, containing all of its subsets. Then a function $f : 2^E \to \mathbb{R}_{\geq 0}$ is *submodular* if it possesses the property of diminishing returns: For all $A \subseteq B \subseteq S$, and for all $x \in S \setminus B$ we have

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

We refer to $f(A \cup \{x\}) - f(A)$ as the *marginal reward* of x given A, and denote it by $\Delta(x|A)$. In addition to submodularity, we will consider functions that possess two further properties:

(i) *Monotonicity:* For all $A \subseteq B \subseteq E$, we have $f(B) \geq f(A)$; and

(ii) *Normalization:* $f(\emptyset) = 0$.

Given a normalized submodular function f defined over a base set E containing $|E| = n$ elements, along with a positive integer $k < n$, the submodular maximization problem can be stated as

$$\max_{S \subset E} \; f(S) \tag{1}$$

$$\text{subject to} \quad |S| \leq k. \tag{2}$$

It is well known that this optimization problem is NP-hard. It contains as a special case several well known combinatorial optimization problems including Max Cut and Facility Location. Even when the function is monotone, the problem is still NP-hard, and contains as a special case Max k-Cover.

When the submodular function is normalized and monotone, a simple greedy algorithm, in which a solution is built incrementally by adding the element x to S that maximizes $\Delta(x|S)$ yields a solution that is within a multiplicative factor of $(1 - 1/\mathrm{e}) \approx 0.63$ of the optimal. In addition, it is shown in [22] that Max k-Cover cannot be approximated to a factor better than $(1 - 1/\mathrm{e})$, unless P = NP. This implies that the greedy algorithm provides the best possible approximation.

In place of the constraint (2), one can use a more general form of constraint known as a matroid. The constraint in (2) defines the *uniform matroid* in which S must belong to $\{A \subseteq E \; : \; |A| \leq k\}$. Another commonly used matroid constraint in optimization is the *partition matroid* defined as follows: Let E be a base set of elements, and partition E into n disjoint sets $E_1, \ldots, E_n$ and let $k_1, \ldots, k_n$ be positive integers. Then allowable sets are members of $\{A \subseteq E \; : \; |A \cap E_i| \leq k_i\}$. It has been shown [14] that maximizing a normalized and monotone submodular function subject to a matroid constraint can be approximated to within $1/2$ using the simple greedy algorithm.

III. Problem Statement

We begin by introducing a centralized version of the submodular maximization problem studied in this paper.

Consider the following submodular maximization problem. We are given n disjoint sets X_i, $i \in \{1, \ldots, n\}$, $X = \cup_i X_i$, and a submodular monotone and normalized function $f : 2^X \to \mathbb{R}_{\geq 0}$. We wish to solve

$$\begin{aligned} &\max_{S \subseteq X} f(S) \\ &\text{subject to} \\ &|S \cap X_i| \leq 1 \quad \text{for each } i \in \{1, \ldots, n\}. \end{aligned} \tag{3}$$

This is a submodular maximization problem over a partition matroid, and thus from Section II, the optimal solution can be approximated to within a multiplicative factor of $1/2$ using the simple greedy algorithm [14].

A. Distributed Optimization Problem

In this paper we consider a *collaborative* submodular analogue of (3).

Problem 3.1 (Distributed submodular maximization): We are given n agents, or players $V = \{1, \ldots, n\}$. Each agent has an *action set* of strategies X_i, and must choose one strategy $x_i \in X_i$. We define $X = \cup_i X_i$ and are given a normalized and monotone increasing submodular function $f : 2^X \to \mathbb{R}_{\geq 0}$. The goal is for the agents to each choose an $x_i \in X_i$ such that $f(x_1, \ldots, x_n)$ is maximized.

We consider the case where agents choose their strategies in a sequential manner, starting with agent 1 and ending with agent n. Each agent $i \in V$ has access to a subset of the strategies that have been chosen by agents $\{1, \ldots, i-1\}$. This information is encoded in a directed acyclic graph (DAG) $\mathcal{G} = (V, E)$ where all edges $(i, j) \in E$ satisfy $i < j$. Note that every DAG can be topologically sorted, and thus there always exists a labeling of the vertices of $\mathcal{G}$ in which $i < j$ for all $(i, j) \in E$. A *complete DAG* is a DAG for which no edge can be added without creating a cycle. A *clique* in $\mathcal{G}$

is a subgraph of $\mathcal{G}$ that is complete. The *clique number* of a DAG $\mathcal{G}$ is the number of vertices in its largest clique, and is denoted $\omega(\mathcal{G})$.

We define the in-neighbors of agent i as

$$\mathcal{N}(i) = \{j \in V \mid (j, i) \in E\},$$

and thus the information available to agent i when choosing its strategy is

$$X_{\text{in}}(i) = \{x_j \mid j \in \mathcal{N}(i)\}.$$

In this paper we study the performance of the greedy algorithm in which agent i chooses its strategy x_i to maximize its marginal reward relative to its limited information $X_{\text{in}}(i)$:

$$x_i = \text{argmax}_{x \in X_i} \Delta\big(x \mid X_{\text{in}}(i)\big). \tag{4}$$

In the case where multiple strategies have equal marginal reward, the greedy algorithm selects one maximizer arbitrarily.

Given a set of strategies $\{x_1, \dots, x_n\}$ for the n agents, the overall objective can be written as a sum of marginal rewards as

$$f(x_1, \dots, x_n) = \sum_{i=1}^{n} \Delta(x_i \mid \{x_1, \dots, x_{i-1}\}), \tag{5}$$

where $\{x_1, \dots, x_0\}$ is defined to be the empty set.

Note that while x_i that maximizes its marginal reward relative to $X_{\text{in}} \subseteq \{x_1, \dots, x_{i-1}\}$, its contribution to the overall objective value is given by the marginal reward relative to $\{x_1, \dots, x_{i-1}\}$. The main goal of this paper is to investigate how this lack of information affects the performance of the greedy algorithm. It is worth mentioning that a version of (4) with synchronous updates can also be analyzed in a similar fashion. Moreover, the assumption on the strategy sets being disjoint can be removed, and all of the following results continue to apply; due to space constraints we will address this issue at length elsewhere.

IV. The Sequential Distributed Greedy Algorithm: Lower Bound

Throughout this section, we assume that the strategy sets are disjoint. The agents take their decisions sequentially in increasing order according to their index. We start with a scenario where the agents *do not observe* the strategies of the agents that have taken action prior to them (see Figure 1(a)). The decision of Agent $i \in V$ is then

$$x_i = \text{argmax}_{x \in X_i} \Delta(x|\emptyset).$$

Suppose that $\{x_1^*, x_2^*, \dots, x_n^*\}$ be the solution of (3), where $x_i^* \in X_i$ for each $i \in \{1, \dots, n\}$. We have that

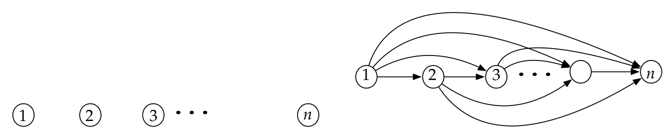
$$f(x_1, \dots, x_n) \geq f(x_i) \geq f(x_i^*),$$

for all $i \in \{1, \dots, n\}$, i.e.,

$$f(x_1, \dots, x_n) \geq \frac{1}{n} \sum_{i=1}^{n} f(x_i^*) \geq \frac{1}{n} f(x_1^*, \dots, x_n^*).$$

It is easy to observe that this lower bound is tight and cannot be improved. A natural problem is hence to investigate if this lower bound on the performance can be improved when the agents can observe (perhaps partially) the decisions of the preceding agents before making decisions. Consider the scenario where agent i observes the decision of all agents in the set $\mathcal{I}_i = \{1, \dots, i-1\}$. The information that each agent has access to before taking its decision is best represented by Figure 1(b). In this case, the decision for agent i from (4) becomes

$$x_i = \text{argmax}_{x \in X_i} \Delta(x|\{x_1, \dots, x_{i-1}\}). \tag{6}$$



(a) (b)

Fig. 1. (a) shows a scenario with no observation and (b) shows a scenario where each agents observes the decisions of all the preceding agents.

The performance of the strategy proposed by (6) can be deduced from the so-called "local greedy algorithms", which is studied in the classical paper [15, Theorem 4.1]. The proof presented in [15] relies on a clever use of linear programming. Here we obtain this result using a different technique, which relies on the next result.

Lemma 4.1: Consider Problem 3.1 and let $(x_1^*, x_2^*, \dots, x_n^*)$ be an optimal solution. Suppose that players sequentially update their strategies $(x_1, \dots, x_n)$ according to (6). Then, for all $1 \leq k \leq n$, we have that

$$f(x_1, \dots, x_k) \geq f(x_1^*, \dots, x_k^*) - f(x_1, \dots, x_{k-1}). \tag{7}$$

Using Lemma 4.1, we can obtain the following bound on the performance of the local greddy algorithm.

Theorem 4.2: Consider Problem 3.1 and let $(x_1^*, x_2^*, \dots, x_n^*)$ be an optimal solution. Suppose that players sequentially update their strategies $(x_1, \dots, x_n)$ according to (6). Then

$$f(x_1, \dots, x_n) \geq \frac{1}{2} f(x_1^*, \dots, x_n^*).$$

We next present a consequence of this result which holds for any general directed acyclic graph.

Corollary 4.3: Consider Problem 3.1 and let $(x_1^*, x_2^*, \dots, x_n^*)$ be an optimal solution. Suppose that players sequentially update their strategies $(x_1, \dots, x_n)$ according to (4) on a directed acyclic graph $\mathcal{G}$. Then,

$$f(x_1, \dots, x_n) \geq \frac{1}{(n - \omega(\mathcal{G})) + 2} f(x_1^*, \dots, x_n^*),$$

where $\omega(\mathcal{G})$ is the clique number of $\mathcal{G}$.
The proof of this result follows from Theorem 4.2, and the fact that the remaining $n - \omega(\mathcal{G})$ agents cannot have a performance worst than $\frac{1}{n - \omega(\mathcal{G})}$ of their optimal.

Note that if we take a complete DAG and delete a single edge, it's clique number reduces from n to $n-1$, and from Corollary 4.3, our lower bound decreases from $1/2$ to $1/3$. The reason for this is that we have not made any assumptions on the relative contribution of each player to the total reward.

The one agent that is removed from the clique may contribute more to the total reward than all $n-1$ other agents combined.

A. Interconnected Cliques of Full Information

In general, it is not clear how the availability or lack of information about the decisions of other agents influences the lower bound on the performance of local greedy algorithms. In this sub-section, we extend the result obtained in Theorem 4.2 to a scenario with multiple cliques, where the agent's in each clique have access to the decision of the last agent in each the last clique that takes decision prior to them. We present this result next.

Theorem 4.4: Consider Problem 3.1 and let $(x_1^*, x_2^*, \ldots, x_n^*)$ be an optimal solution. Suppose that first players $\{1, \ldots, m\}$ and players in $\{m+1, \ldots, n\}$ form two cliques with the only observation from the first clique available to the players of the second clique being the choice of player m. Suppose that players sequentially update their strategies $(x_1, \ldots, x_n)$ according to (4). Then we have that

$$f(x_m, \ldots, x_n) \geq \frac{1}{2} f(x_m^*, \ldots, x_n^*) + \frac{1}{2}(f(x_m, x_{m+1}) - f(x_m^*, x_{m+1}^*)).$$

Moreover, we have that

$$f(x_1, \ldots, x_n) \geq \frac{1}{4} f(x_1^*, \ldots, x_n^*).$$

Remark 4.5 (Comparison with isolated cliques): Suppose that there is no communications between agents in the two cliques. Then, since

$$f(x_1, \ldots, x_m) \geq \frac{1}{2} f(x_1^*, \ldots, x_m^*), \quad \text{and}$$
$$f(x_{m+1}, \ldots, x_n) \geq \frac{1}{2} f(x_{m+1}^*, \ldots, x_n^*),$$

using submodularity and monotonicity, we conclude that $f(x_1, \ldots, x_n) \geq \frac{1}{4} f(x_1^*, \ldots, x_n^*)$. In this sense, Theorem 4.4 captures a scenario where the overall performance can only get enhanced by having access to more information; in fact, as shown in (**??**), the performance can be improved by $\frac{1}{4}(f(x_m, x_{m+1}) + f(x_m^*) - f(x_m^*, x_{m+1}^*)) \geq 0$. •

The result presented in Theorem 4.4 is extendable to the case of more than two cliques, with the tail agent in each clique broadcasting its decision to the next clique. The proof is similar and will not be provided here.

Theorem 4.6: Consider Problem 3.1 and let $(x_1^*, x_2^*, \ldots, x_n^*)$ be an optimal solution. Suppose that agents are partitioned into $\kappa \in \mathbb{Z}_{\geq 2}$ cliques as

$$\{1, \ldots, m_1\}, \{m_1+1, \ldots, m_2\}, \cdots \{1, \ldots, m_\kappa\},$$

with the only information available from other cliques to the agent's in $(i+1)$'s clique being the choice of agent m_i, $i \in \{1, \ldots, \kappa\}$. Suppose that players sequentially update their strategies $(x_1, \ldots, x_n)$ according to (4). Then

$$f(x_1, \ldots, x_n) \geq \frac{1}{2^\kappa} f(x_1^*, \ldots, x_n^*).$$

As noted in Remark **??**, the results of this section are extendable to the case where the information exchange is synchronous. We postpone a detailed discussion of synchronous updates to a future work.

V. Upper bounds using colouring

In this section we determine upper bounds on the performance of the sequential greedy algorithm. We study two cases: 1) a graph dependent submodular function that exploits information on graph topology; and 2) a global submodular function that provides bounds for all graphs. We will see that these two cases are essentially equivalent to computing an optimal and a greedy coloring [23] of the graph, respectively. The second case also allows us to extract graph properties that limit performance.

A. Background on Graph Coloring

Given a graph $G = (V, E)$ with $|V| = n$, a *coloring* is a function $c : V \to \{1, \ldots, n\}$ such that for every $(u, v) \in V$, we have $c(u) \neq f(v)$. The *vertex-coloring problem* is to find a coloring $c : V \to \{1, \ldots, k\}$ such that k is minimized. That is, the goal is to color the vertices of the graph with the minimum number of colors such that no pair of adjacent vertices have the same color. The optimum value of k is called the *chromatic number* of the graph G and is often denoted by $\chi(G)$. The problems of computing a minimum coloring or equivalently of determining the chromatic number of a graph are NP-hard [23].

B. Upper Bound via Adversarial Choice of Function

In this sub-section we consider the following problem: Given a graph $\mathcal{G} = (V, E)$, can we construct a submodular function $f_\mathcal{G}$, exploiting information of the graph for which we can upper bound the performance of greedy algorithm? Given a graph $\mathcal{G} = (V, E)$, we construct action sets and a submodular function as follows. Let $c : V \to \{1, \ldots, k\}$ be an optimal coloring of $\mathcal{G}$ and where $\chi(\mathcal{G}) = k$ is the chromatic number of $\mathcal{G}$. Given a coloring, we define the set of vertices of color $\ell \leq k$ as

$$V_\ell = \{v \in V \mid c(v) = \ell\}.$$

We use this coloring to construct strategy sets X_i for each agent $i \in V$ and the submodular function $f : 2^X \to \mathbb{R}_{\geq 0}$. For each agent i, we define

$$X_i = \{a_i, b_i\},$$

and let $X = \cup_i X_i$. We define the function f through its marginal rewards. To that end, let $S \subset X$ be a set such that $|S \cap X_i| \leq 1$ containing decision for a subset of the n agents in V. Then, the marginal reward of a_i is defined as

$$\Delta(a_i|S) = \begin{cases} 0 & \text{if } a_j \in S \text{ for some } j \in V_{c(i)}, \\ 1 & \text{otherwise.} \end{cases}$$

The marginal reward of b_i is defined is $\Delta(b_i|S) = 1$, for all $i \in V$. Then, given a set $S = \{s_1, \ldots, s_m\} \subset X$ with $|S \cap X_i| \leq 1$, we define

$$f(S) = \sum_{i=1}^{m} \Delta\big(s_i|\{s_1, \ldots, s_{i-1}\}\big). \tag{8}$$

Remark 5.1 (Non-disjoint Action Sets): Note that we can equivalently define this function over non-disjoint action sets. Let $X = \{a, b_1, \ldots, b_n\}$ and define $X_i = \{a, b_i\}$. Then given a set $S \subset X$, we define $f(S) = |S|$. Notice that this function is modular. Given a tuple containing agent strategies $(x_1, \ldots, x_n)$, we define the set of unique strategies as $S = \{x_1, \ldots, x_n\}$ and evaluate $f(S)$ as the reward. This definition is equivalent to the disjoint definition above as discussed in Section **??**. •

With this function we obtain the following result.

Proposition 5.2: Consider Problem 3.1, where f is given by (8), and let $(x_1^*, x_2^*, \ldots, x_n^*)$ be an optimal solution. Suppose that players sequentially update their strategies $(x_1, \ldots, x_n)$ according to (4). Then,

$$f(x_1, \ldots, x_n) \leq \frac{\chi(G)}{n} f(x_1^*, \ldots, x_n^*).$$

C. Upper Bound via Universal Function

The upper bound in the previous section required a submodular function f that depended on the graph. In the following, we propose a single function, independent of the graph topology, and analyze its performance for all graphs. This allows us to state a few simple properties of graphs that limit its performance.

The base set of the function is $X = \{e_1, \ldots, e_m\}$ where $m \geq n$ and we let $X_i = X$ for each agent i. Given a choice $x_i \in X_i$ for each agent i, the value of the submodular function is

$$f(x_1, \ldots, x_n) = | \cup_i \{x_i\}|. \qquad (9)$$

That is, the value is given by the number of unique elements of X chosen by the agents. Clearly, an optimal solution is any one in which each agent chooses a different element from X, yielding a value of n. Under the greedy algorithm, agent i will compute its marginal reward as

$$\Delta(e_j \mid X_{\text{in}}(i)) = \begin{cases} 1 & \text{if } e_j \notin X_{\text{in}}(i) \\ 0 & \text{if } e_j \in X_{\text{in}}(i), \end{cases}$$

and thus agent i will choose any strategy e_j such that $e_j \notin X_{\text{in}}(i)$. Suppose that each agent i breaks ties by choosing the strategy with lowest index. Then, we can write the greedy choice for agent i as

$$x_i = \min\{e_j \in \{1, \ldots, m\} \mid e_j \notin X_{\text{in}}(i)\}. \qquad (10)$$

Our next result relates the performance of the greedy algorithm on this function to properties of the underlying graph.

Proposition 5.3: Consider the submodular function in (9) and any graph $\mathcal{G} = (V, E)$. If the greedy algorithm finds a solution within k/n of the optimal for some $k > 0$ then each of the following properties hold:

(i) there is a vertex in $\mathcal{G}$ with in-degree of at least $k - 1$;
(ii) for each $\ell \in \{1, \ldots, k\}$ there are at least ℓ agents with in-degree of at least $k - \ell$;
(iii) the number of edges in $\mathcal{G}$ is at least $k(k-1)/2$.

Notice that the sequence of strategy choices in (10) provides a simple and efficient algorithm for computing a performance upper bound for a given graph. For completeness, we give the details in Algorithm 1.

Algorithm 1 Greedy Algorithm Upper Bound

Input: A directed acyclic graph $\mathcal{G} = (V, E)$.
Output: An upper bound on the approximation ratio of greedy algorithm on $\mathcal{G}$.
1: Topologically sort the vertices V
2: **for** each $v \in V$:
3: Set $\texttt{value}[v] = 0$
4: **for** each $v \in V$ in topological order :
5: Set $\texttt{value}[v]$ to smallest integer k such that for each $u \in \mathcal{N}(i)$, $\texttt{value}[u] \neq k$.
6: **return** $\frac{1}{|V|} \max_{v \in V} \texttt{value}[v]$

Complexity of Algorithm 1: The complexity of Algorithm 1 is $O(|V| + |E|)$. The topological sort can be performed in $O(|V| + |E|)$ time. The only detail is to implement line 5: in linear time, which essentially computes the smallest element not in an array. This can be done using two passes through the array of in-neighbor values. In the first pass, we populate a Boolean array of length $|\mathcal{N}(i)|$. All entries of the array are initialized to `false`, and the jth entry is then reset to `true` if and only if there is a vertex $u \in \mathcal{N}(i)$ with $\texttt{value}[u] = j$. In the second pass, we scan the Boolean array for the first false entry. This is the smallest value that is not chosen by an in-neighbor. Thus, the total computation time for the for-loop in line 4: is $O(|E|)$.

D. Gap Between Adversarial and Universal Upper Bounds

The strategies chosen by each agent for the submodular function in (9) provides a coloring of the graph $\mathcal{G}$. That is, the vertices $V_\ell := \{i \in V \mid x_i = e_\ell\}$ are those colored with color ℓ. By construction, there are no edges between vertices of the same color. This implies that the adversarial upper bound is tighter than the universal upper bound, which utilizes the minimum number of colors.

A key question is how large the gap can be between the two upper bounds. In general it can be arbitrarily large. To see this, consider the following bipartite directed graph $\mathcal{G} = (V, E)$ consisting of $n = 2m$ vertices, where $V = \{u_1, \ldots, u_m\} \cup \{w_1, \ldots, w_m\}$. The graph contains all edges (u_i, v_j) such that $i < j$ and all edges (v_i, u_j) such that $i < j$ along with the edge (u_m, w_m). An example is shown in Figure 2. For this graph, under Algorithm 1, the vertex w_m chooses strategy e_{m+1}. Thus, we have that

(i) the chromatic number of $\mathcal{G}$ is 2, and thus the adversarial upper bound is $2/n$.
(ii) Algorithm 1 returns $1/2 + 1/(2m)$.

However, the advantage of Algorithm 1 is that it is a simple linear-time algorithm for computing an upper bound on achievable performance of a given graph topology. In contrast, the adversarial function of Section V-B provides a much tighter upper bound, but requires solving an NP-hard problem. An interesting connection to note is that Algorithm 1 is essentially computing a greedy coloring of

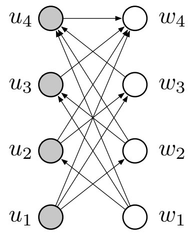


Fig. 2. A bipartite graph on eight vertices. The graph is colored using just two colors, and its chromatic number is two. Under (10), vertex w_4 will select strategy e_5, and thus Algorithm 1 will output $5/8$.

$\mathcal{G}$, where vertices are considered in topological order. It is known that in general a greedy coloring does not provide a constant factor approximation to the minimum coloring [23].

E. Comparison of Lower and Upper Bounds

In Section IV we provided lower bounds on the performance of greedy algorithm (6) for any monotone, normalized, submodular function. This consisted of results for several graph topologies, and a general result based on the clique number of the graph. Table I compares these lower bounds with the upper bounds obtained from Proposition 5.2 and Algorithm 1. Given a graph topology, the lower bound provides a minimum performance guarantee for all submodular functions. In constrast, the upper bounds provide limitations on performance for a specific (worst-case) submodular function.

Graph	Lower	$\chi(\mathcal{G})$ Upper	Alg. 1 Upper
Empty	$1/n$	$1/n$	$1/n$
Complete acyclic	$1/2$	1	1
Interconnected cliques	$1/2^\kappa$	$1/\kappa + 1/n$	$1/\kappa + 1/n$
General DAG	$\frac{1}{n-\omega(\mathcal{G})+2}$	$\chi(\mathcal{G})/n$	Alg. 1

TABLE I

COMPARISON BETWEEN THE LOWER BOUND IN SECTION IV AND THE TWO UPPER BOUNDS FOR THREE GRAPH TOPOLOGIES.

The entires in Table I for the interconnected cliques graph assume κ cliques, each of size n/κ.

VI. CONCLUSIONS AND FUTURE WORK

We have proposed a distributed setting for a class of submodular maximization problems under matroid constraints, where the strategy set is partitioned into private strategy sets assigned to a group of agents. We have investigated the limitations that the lack of information about the actions of other agents can impose on the performance of local greedy algorithms. Investigating the applications of distributed submodular optimization, obtaining tighter upper bounds for more homogeneous classes of submodular functions, and investigating under what additional assumptions on the submodular functions one can generalize the lower bounds provided in this paper are interesting avenues of future research.

REFERENCES

[1] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Transactions on Automatic Control*, vol. 31, no. 9, pp. 803–812, 1986.

[2] M. Rabbat and R. Nowak, "Distributed optimization in sensor networks," in *Symposium on Information Processing of Sensor Networks*, (Berkeley, CA), pp. 20–27, Apr. 2004.

[3] L. Xiao and S. Boyd, "Optimal scaling of a gradient method for distributed resource allocation," *Journal of Optimization Theory & Applications*, vol. 129, no. 3, pp. 469–488, 2006.

[4] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[5] P. Wan and M. D. Lemmon, "Event-triggered distributed optimization in sensor networks," in *Symposium on Information Processing of Sensor Networks*, (San Francisco, CA), pp. 49–60, 2009.

[6] A. Nedic, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 922–938, 2010.

[7] B. Johansson, M. Rabi, and M. Johansson, "A randomized incremental subgradient method for distributed optimization in networked systems," *SIAM Journal on Control and Optimization*, vol. 20, no. 3, pp. 1157–1170, 2009.

[8] B. Gharesifard and J. Cortés, "Distributed continuous-time convex optimization on weight-balanced digraphs," *IEEE Transactions on Automatic Control*, vol. 59, no. 3, pp. 781–786, 2014.

[9] A. Nedic and A. Olshevsky, "Distributed optimization over time-varying directed graphs," *IEEE Transactions on Automatic Control*, vol. 60, no. 3, pp. 601–615, 2015.

[10] J. R. Marden, "The role of information in multiagent coordination," in *IEEE Conf. on Decision and Control*, pp. 445–450, 2014.

[11] S. T. Jawaid and S. L. Smith, "On the submodularity of sensor scheduling for estimation of linear dynamical systems," in *American Control Conference*, pp. 4139–4144, 2014.

[12] S. T. Jawaid and S. L. Smith, "Informative path planning as a maximum traveling salesman problem with submodular rewards," *Discrete Applied Mathematics*, vol. 186, pp. 112–127, 2015.

[13] A. Krause and D. Golovin, "Submodular function maximization," *Tractability: Practical Approaches to Hard Problems*, vol. 3, p. 19, 2012.

[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] M. L. Fisher, G. L. Nemhauser, and L. A. Wolsey, "An analysis of approximations for maximizing submodular set functions-II," in *Mathematical Programming Study 8*, pp. 73–87, North-Holland Publishing Company, 1978.

[16] L. Lovász, "Submodular functions and convexity," in *Mathematical Programming The State of the Art*, pp. 235–257, Springer, 1983.

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

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

[19] 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*, pp. 2049–2057, 2013.

[20] B. Mirzasoleiman, A. Karbasi, R. Sarkar, and A. Krause, "Distributed submodular maximization," *arXiv preprint arXiv:1411.0541*, 2014.

[21] P. Stobbe, *Convex analysis for minimizing and learning submodular set functions*. PhD thesis, California Institute of Technology, 2013. Ph.D. thesis.

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

[23] B. Korte and J. Vygen, *Combinatorial Optimization: Theory and Algorithms*, vol. 21 of *Algorithmics and Combinatorics*. Springer, 4 ed., 2007.

[24] B. Bollobás and P. Erdös, "Cliques in random graphs," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 80, no. 03, pp. 419–427, 1976.