A Programming Approach for Worst-case Studies in Distributed Submodular Maximization

Andrew Downie, Bahman Gharesifard, and Stephen L. Smith

Abstract— **We present a method to realize a submodular function as a vector in the feasible region of a set of linear constraints. We utilize this representation to formulate a linear program to find worst-case functions for the greedy strategy in distributed submodular maximization. This construction provides insight into the structure of worst-case functions, and enables us to better understand how the team's performance is affected by changing the network structure.**

I. Introduction

A submodular function is a set function that exhibits diminishing returns. That is, the contribution of an element to the total value of a set decreases as the set gets bigger. In submodular maximization, we seek to find a set that maximizes a submodular function subject to some constraints (for example, on the number of elements in the set). Interest in submodular maximization has grown rapidly due to the fact that it models a variety of important problems. Some real world domains where submodular maximization has application is in controls [1], [2], [3], [4], path planning [5], [6], [7], sensor placement and sensor scheduling [8], [9], [10], [11], [12], [13], [14]. While submodular maximization is NP-hard [15], in almost all of these problems, a simple greedy strategy provides strong approximation guarantees.

A large portion of the theoretical research in the domain of submodular maximization is concerned with providing approximation guarantees. It is common that the greedy strategies for maximization perform better in practice than the theoretical guarantees. With this in mind, we are interested in answering the following question: Is it possible to find a submodular function f such that the solution produced by the greedy strategy exhibits its worst-case approximation ratio? It is typically challenging to find such examples, and one is interested to investigate if they can be found programmatically.

In this work, we establish a connection between submodular functions and feasible regions of linear constraints. We use linear programming to find these worst-case function examples. The idea of utilizing linear programming has some parallels to a proof technique used in the foundational paper of Nemhauser et. al [16], where for a given function, the authors upper bound the optimal value of the problem and then pose a linear program for lower bounding the value of the greedy solution. In our work, we are interested in generalizing these methods as well as finding the particular functions that exhibit the worst-case values. Such worst-case scenarios play a major role in previous work related on the role of information in distributed submodular optimization, including the ones presented in [17], [18], [19], [20]. To elaborate further on the distributed settings we have in mind, consider a setting where a team of agents want to collaboratively maximize a submodular objective function. Each agent must select an action from their own set of available actions. The agents make choices based on a subset of actions selected by other agents. Since the agents cannot observe all of the other agents actions they are forced to make decisions under partial information. The agents make their choices by greedily maximizing their own marginal objective value given the information available to them.

Statement of Contributions: We provide a representation of a submodular function as a solution of a linear program, where given a base set of N elements, we represent f as a 2^N dimensional vector, and we enforce submodularity by imposing linear constraints on the vector. We then consider a distributed submodular maximization problem, where agents have partial information about the decisions of other agents, over both the *partition matroid*. Given a team of agents and a graph describing the agents' available information, we formulate a linear program that provides a worst-case submodular function, in terms of approximation performance. This allows us to characterize the size of the domain of the worst-case functions.

II. Preliminaries

We begin with some basic preliminaries. Let X be a base set of elements, and 2^X be the power set of X. A set function $f : 2^X \to \mathbb{R}$ is submodular if for all sets $S, T \subseteq X$, the following holds:

$$f(S) + f(T) \geq f(S \cap T) + f(S \cup T). \quad (1)$$

We denote the marginal gain of adding $x \in X$ to a set $S \subseteq X$ by

$$f(x|S) := f(\{x\} \cup S) - f(S).$$

In addition to submodularity, throughout this paper, we assume the following additional properties

1) Monotonicity: For all $S \subseteq X$, $f(x|S) \geq 0$,
2) Normalization: $f(\emptyset) = 0$.

Definition II.1 (Matroid)**.** Let X be a finite set and $\mathcal{I}$ a non-empty collection of subsets of X called the independent sets. The system $M = (X, \mathcal{I})$ is called a matroid if:

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC)

Andrew Downie and Stephen L. Smith are with are with the Electrical and Computer Engineering at the University of Waterloo, Waterloo, ON, Canada {adownie,stephen.smith}@uwaterloo.ca

Bahman Gharesifard is with the Department of Electrical and Computer Engineering at the University of California, Los Angeles gharesifard@ucla.edu

1) If $S \subseteq T \subseteq X$ and $T \in \mathcal{I}$ then $S \in \mathcal{I}$.
2) If for all $S, T \in \mathcal{I}$ and $|S| < |T|$ then there exists $x \in T \backslash S$ such that $S \cup \{x\} \in \mathcal{I}$.

We are interested in a particular matriod called the partition matriod where X is split into $X_1 \dots, X_n$ disjoint sets, and $\mathcal{I} = \{S \subseteq X \mid |S \cap X_i| \leq 1 \text{ for all } i \in \{1, \dots, n\}\}$

III. Linear Programming Representation

We start this section by introducing a method to realize submodular functions as vectors in the feasible region of a set of linear constraints. This bears resemblance with the techniques used in the original formulation of the submodular maximization problem in the classical work of [16]. As we show in Section IV-C this representation will enable us to pose an optimization problem that generates worst-case functions for greedy optimization strategies.

To this end, let X be a base set of elements. We represent a set function defined on X as a 2^N dimensional real-valued vector, where $N = |X|$. Let $v \in \mathbb{R}^{2^N}$ be a vector (or lookup table) where each component of v gives the function value for a corresponding subset $S \subseteq X$; we denote this component by v_S. We assume that there is a fixed ordering of subsets, in that the indexing of subsets is fixed i.e., for any two vectors $v, \hat{v} \in \mathbb{R}^{2^N}$, the values of v_S and $\hat{v}_S$ are located at the same index for the two vectors. Given $v \in \mathbb{R}^{2^N}$, we define a set function $f : 2^X \to \mathbb{R}$ by

$$f(S) = v_S.$$

We can enforce properties on the function f by imposing constraints on v.

Example III.1. Consider the base set $X = \{x_1, x_2, x_3\}$. The vector representation of a function $f : 2^X \to \mathbb{R}$ is a vector $v \in \mathbb{R}^8$ with the following structure:

$$v = \begin{bmatrix} f(\emptyset) \\ f(x_1) \\ f(x_2) \\ f(x_3) \\ f(x_1, x_2) \\ f(x_1, x_3) \\ f(x_2, x_3) \\ f(x_1, x_2, x_3) \end{bmatrix}$$

Note that a vector $v \in \mathbb{R}^{2^N}$, uniquely defines a set function $f_v : 2^X \to \mathbb{R}$.

We now define a matrix $M_{\text{submodular}}$, with the property that if $M_{\text{submodular}} v \geq 0$ then, the function f defined by v is submodular. We now rewrite (1), in terms of the components of v as

$$v_S + v_T - v_{S \cap T} - v_{S \cup T} \geq 0. \tag{2}$$

Let $M_{\text{submodular}} \in \mathbb{R}^{2^N(2^N-1) \times 2^N}$, where each row corresponds to the constraint in (2) for a pair of subsets S and T. We now define another matrix to ensure the function represented by v is also monotone.

Example III.2. An example row of $M_{\text{submodular}}$ for function in Example III.1 with $S = \{x_1, x_2\}$ and $T = \{x_3\}$ is

$$\begin{bmatrix} -1 & 0 & 0 & 1 & 1 & 0 & 0 & -1 \end{bmatrix} v \geq 0.$$

We now define another matrix to ensure the function represented by v is also monotone. For f to be monotone we need to ensure that

$$f(x|S) \geq 0,$$

for all $x \in X$ and $S \subseteq X$. we can enforce monotonicity on v by adding N more constraints. In particular, if f is submodular then $f(x|S) \geq f(x|X \backslash \{x\})$ for $S \subseteq X \backslash \{x\}$. Hence, enforcing $f(x|X \backslash \{x\}) \geq 0$ implies $f(x|S) \geq 0$. Written in terms of the components v, we impose the condition

$$v_X - v_{X \backslash \{x\}} \geq 0. \tag{3}$$

Let $M_{\text{monotone}} \in \mathbb{R}^{N \times 2^N}$, where each row encodes the monotonicity constraint imposed by x in (3) for all $x \in X$. If $M_{\text{submodular}} v \geq 0$ and $M_{\text{monotone}} v \geq 0$ hold, the function represented by v will be both monotone and submodular. Finally to ensure that f is normalized we simply impose an equality constraint $v_\emptyset = 0$.

A key fact about these constraints is that they are all linear inequalities in terms of the components of v. We formulate a linear program in order to search for monotone, normalized and submodular functions while imposing other desirable linear constraints:

$$\begin{aligned} &\max_{v \in \mathbb{R}^{2^N}} c^T v, \\ &\text{s.t.} \begin{bmatrix} M_{\text{submodular}} \\ M_{\text{monotone}} \end{bmatrix} v \geq 0, \\ &v_\emptyset = 0, \quad \text{and} \quad Mv \geq b, \end{aligned}$$

where $c \in \mathbb{R}^{2^N}$ is a general cost vector, $M \in \mathbb{R}^{l \times 2^n}$ and $b \in \mathbb{R}^l$. Here, M and b are general constraints that can be used to enforce additional properties on the submodular function produced by the optimal solution v. As an example in Section V, we specify c, M and b to construct a linear program that produces performance guarantees for the adapted greedy strategy for distributed submodular maximization.

IV. Distributed Submodular Maximization

A. Problem Definition

We now introduce the distributed submodular maximization problem [17] and how we can apply our linear programming approach. Suppose we are given n agents $V = \{1, \dots, n\}$. We consider the scenario where the agents select their actions sequentially. To state this formally, suppose that each agent i has access to an action set X_i and must choose one action $x_i \in X_i$. The agents follow a greedy strategy and we want study the impact of information on their performance. Each agent $i \in V$ has access to a subset of decisions chosen by agents $\{1, \dots, i-1\}$, encoded by a directed acyclic graph (DAG) $\mathcal{G} = (V, E)$, where, there is an edge $(i, j) \in E$ if agent j has access to the action of agent i. We refer to this graph as the agents' communication graph.

The in-neighbor set of agent i in $\mathcal{G}$ is defined as

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

The information available to this agent is given by

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

We assume the agents select their actions by greedily maximizing their own marginal return given the information available to them, i.e.,

$$x_i \in \arg\max_{x\in X_i} f(x|X_{\text{in}}(i,\mathcal{G})). \tag{4}$$

We denote a greedy solution by $S_{\mathcal{G}} = \{x_1,\dots,x_n\}$ where each x_i satisfies (4), and the set of all greedy solutions by $\mathbb{S}_{\mathcal{G}}$.

We study the problem described in [17], [18], [20], defined as follows:

$$\begin{aligned} &\max_{S\subseteq X, |S|\le n} f(S) \\ &\text{s.t. } |S\cap X_i| \le 1 \text{ for } i\in\{1,\dots,n\}, \end{aligned} \tag{5}$$

where $X = \cup_{i\in n} X_i$ and each X_i is disjoint. This is an instance of maximizing a submodular function over the *partition matroid*.

Suppose we have a normalized, monotone and submodular function f, a group of agents, a DAG $\mathcal{G}$, and the greedy strategy given by (4). We want to study the worst-case suboptimality of the greedy strategy (4) for the problem defined in (5). Note that this will depend on the information structure $\mathcal{G}$, and thus we want to study how the performance depends on $\mathcal{G}$.

B. Performance Guarantees

Following [18], we define the competitive ratio for a normalized monotone submodular function f and a DAG $\mathcal{G} = (V,E)$ as

$$\gamma(f,X,\mathcal{G}) = \min_{S_{\mathcal{G}}\in\mathbb{S}_{\mathcal{G}}} \frac{f(S_{\mathcal{G}})}{f(OPT)},$$

where $OPT \subseteq X$ is the solution to Problem (5).

Let us denote the worst-case competitive ratio for a given graph $\mathcal{G}$ by

$$\gamma(\mathcal{G}) = \inf_{f,X} \gamma(f,X,\mathcal{G}). \tag{6}$$

Both [17] and [18] provide performance bounds in terms of properties of the graph $\mathcal{G}$. The tightest known bound for this problem is provided in [18] and is in terms of the *fractional independence* number of the graph $\mathcal{G} = (V,E)$.

C. Linear Programming Approach

We will now present how to incorporate the adapted greedy strategy in (4) into our linear programming model. Suppose we are given a DAG $\mathcal{G}$ with n nodes and a set $B = \{x_1,\dots,x_n\}$ that satisfies the partition matroid constraint. We next show that we can add constraints to the linear program such that the feasible set contains all the submodular functions for which B is a greedy solution from (4) i.e, $B\in\mathbb{S}_{\mathcal{G}}$:

To encode the greedy condition, suppose the agents choose $B = \{x_1,\dots,x_n\}$. Then, the information available to agent i is

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

We next search for a $f: 2^X \to \mathbb{R}$ that satisfies

$$f(x_i|X_{\text{in}}(i,\mathcal{G})) \ge f(x|X_{\text{in}}(i,\mathcal{G})) \text{ for all } x\in X_i,$$

for each $x_i \in B$. The latter can be rewritten in terms of the components of a corresponding vector v by

$$v_{\{x_i\}\cup X_{\text{in}}(i,\mathcal{G})} - v_{\{x\}\cup X_{\text{in}}(i,\mathcal{G})} \ge 0 \text{ for all } x \in X_i. \tag{7}$$

Let us define a $M_{\text{greedy},\mathcal{G}} \in \mathbb{R}^{(|X_1|+\cdots+|X_n|)\times 2^N}$, which encodes each of the greedy constraints for all $x_i \in B$. If $M_{\text{greedy},\mathcal{G}} v \ge 0$ then the feasible region of v describes all the set functions where B is a greedy solution.

Example IV.1. Consider the function in Example III.1, two agents with $X_1 = \{x_1\}$ and $X_2 = \{x_2, x_3\}$, and the information graph $\mathcal{G}$ where agent 2 has access to the choice of agent 1. Then, to enforce that x_2 is the greedy choice for agent 2, we have the following row in $M_{\text{greedy},\mathcal{G}}$:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \end{bmatrix} v \ge 0.$$

V. Worst-case functions using linear programs

A. Linear Program Formulation

We begin this section, with a formulation of a linear program whose solution is a function with minimum competitive ratio, out of all the submodular functions where $S_{\mathcal{G}}$ is selected by agents following (4). Given sets $A, B \subseteq X$ that satisfy the partition matroid constraint, i.e., $|A\cap X_i| = 1$ and $|B \cap X_i| = 1$ for all $i\in\{1,\dots,n\}$, the following program produces a submodular function where A is an optimal solution to (5), and B is a greedy solution produced by algorithm (4) with minimum competitive ratio. In particular, we define the following program:

$$\begin{aligned} &\max_{v\in\mathbb{R}^{2^N}} \quad v_A, \\ &\text{s.t.} \quad \begin{bmatrix} M_{\text{submodular}} \\ M_{\text{monotone}} \\ M_{\text{greedy},\mathcal{G}} \end{bmatrix} v \ge 0, \\ &v_\emptyset = 0 \quad \text{and} \quad v_B = \mu. \end{aligned} \tag{8}$$

where the constant μ fixes the value of $f(B)$ for the resulting function. Let v be in the feasible region of the program and f_v be the corresponding function. The program finds the maximum value that the set A can take on given $f(B) = \mu$ with $B \in \mathbb{S}_{\mathcal{G}}$. Since $f(B)$ is fixed, maximizing $f(A)$ will produce the largest value for $f(A)$, and therefore will produce the function with minimum competitive ratio. We will formally prove this result in the next subsection.

Before presenting our main result, we make a remark about on the complexity of solving (8). Using an off-the-shelf black box optimizer, linear program (8) can be solved numerically to produce the worst-case function examples. However, the drawback is that the constraints in the linear programs scale exponentially with the size of X. The encoding of the general linear program requires $\mathcal{O}(2^{|X|}(2^{|X|}-1))$ space using sparse matrix representations. For small enough sets X, black-box optimizers are able to solve the linear programs in a tolerable amount of time.

B. Main Result

Consider the distributed submodular maximization problem with n agents over an information graph $\mathcal{G}$. Let A and B be disjoint sets each of size n. Let $X' = A \cup B$, and define a partition of X' such that $|X'_i \cap A| = 1$ and $|X'_i \cap B| = 1$ for each $i \in \{1, \ldots, n\}$. With this information we present our main result.

Theorem V.1. *Consider A, B, and X' as described above, an information graph $\mathcal{G}$, and let $v^* \in \mathbb{R}^{2^{2n}}$ be a solution to linear program (8) defined using A,B and $\mathcal{G}$ with $\mu = 1$. The function $f_v^* : 2^{X'} \to \mathbb{R}$ corresponding to v^* is a worst-case function for information graph $\mathcal{G}$, i.e.,*

$$\gamma(\mathcal{G}) = \frac{f_{v^*}(B)}{f_{v^*}(A)}.$$

Our proof hinges on the following key lemma.

Lemma V.2. *Let $X_1, \ldots, X_n$ be nonempty disjoint sets with $X = \cup_{i=1}^n X_i$, and let $f : 2^X \to \mathbb{R}$ be a normalized, monotone and submodular function with greedy solution $S_\mathcal{G}$ under a communication graph $\mathcal{G}$ and optimal solution OPT. Let $G = \{g_1, \ldots, g_n\}$, $O = \{o_1, \ldots, o_n\}$, $X' = G \cup O$, and $X'_i = \{g_i, o_i\}$ for all $i \in \{1, \ldots, n\}$. Then there exist a corresponding function $f' : 2^{X'} \to \mathbb{R}$ with the following properties:*

(i) f' is normalized, monotone and submodular;
(ii) O is an optimal solution for f';
(iii) G is a worst-case greedy solution to for f';
(iv) $f'(G) = f(S_\mathcal{G})$;
(v) $f'(O) = f(OPT)$.

The lemma essentially states for every submodular function f defined over an arbitrary action sets $X_1, \ldots, X_n$ and information given an information graph $\mathcal{G}$, there is a corresponding function defined over *disjoint* action sets $X'_1, \ldots, X'_n$ of *size two* that has the identical competitive ratio, with optimal and greedy solutions being disjoint.

Proof of Lemma V.2. Given f we construct f' as follows: Let $S_\mathcal{G} = \{x_1, \ldots, x_n\}$ where $x_i \in X_i$, be a worst-case greedy solution of f i.e. $S_\mathcal{G} \in \arg\min_{S \in \mathbb{S}_\mathcal{G}} f(S)$, let $OPT = \{x_1^*, \ldots, x_n^*\}$ where $x_i^* \in X_i$ for each $i \in \{1, \ldots, n\}$. Since f is arbitrary, $S_\mathcal{G}$ and OPT are not necessarily disjoint.

To begin, we construct the action sets $X'_i = \{o_i, g_i\}$. For each $i \in \{1, \ldots, n\}$, we let $g_i = x_i$. We let $o_i = x_i^*$ if $x_i \neq x_i^*$ or $o_i = \hat{x}_i$ where $\hat{x}_i$ is a copy of x_i^*, otherwise. By construction, each action set contains two elements, and thus the sets $G = \{g_1, \ldots, g_n\}$ and $O = \{o_1, \ldots, o_n\}$ are disjoint.

Next we construct the function f' over $2^{X'}$. Let $p : X' \to X$ where for any $x \in X'$, $p(x)$ returns the corresponding element in X, i.e., $p(g_i) = x_i$ and $p(o_i) = x_i^*$. We then define a function $h : 2^{X'} \to 2^X$ that for any set $S' \subseteq X'$ returns the corresponding set of elements in X as

$$h(S) = \{x \in X | p(x') = x \text{ for some } x' \in S'\}.$$

With this, we define the function $f' : 2^{X'} \to \mathbb{R}$ as

$$f'(S') = f(h(S')).$$

Proof of (i): We now show that f' is normalized, monotone and submodular. This proof closely follows the argument in [17, Section IIID]. Note that $h(X'_i) = X_i$, and by the definition of h, and for any $S', T' \subseteq X'$,

$$h(S') \cup h(T') = h(S' \cup T'), \text{ and } h(S') \cap h(T') = h(S' \cap T').$$

To show that f' is submodular, for any $S', T' \subseteq X'$, we have

$$\begin{aligned} f'(S') + f'(T') &= f\big(h(S')\big) + f\big(h(T')\big) \\ &\geq f\big(h(S') \cup h(T')\big) + f\big(h(S') \cap h(T')\big) \\ &= f\big(h(S' \cup T')\big) + f\big(h(S' \cap T')\big) \\ &= f'(S' \cup T') + f'(S' \cap T') \end{aligned}$$

Therefore, f' is submodular. To see that f' is monotone, for any $S' \subseteq T' \subseteq X'$, we have

$$\begin{aligned} f'(T') &= f(h(T')) \\ &= f\big(h(T' \cap S') \cup h(T' \backslash S')\big) \\ &= f\big(h(T' \cap S')\big) + f\big(h(T' \backslash S') | h(T' \cap S')\big) \\ &\geq f\big(h(T' \cap S')\big) \qquad (9) \\ &= f'(S') \end{aligned}$$

where (9) holds because f is monotone. Therefore f' is monotone. Finally, $f'(\emptyset) = f\big(h(\emptyset)\big) = f(\emptyset) = 0$, and thus f' is normalized.

Proof of (ii): We now show that the optimal value of f' is the same as the optimal value of f, and the solution that achieves the optimal value is O. First note that $h(O) = OPT$ since for each $o_i \in O$, we have $h(o_i) = x_i^*$. Now, let $S' \subseteq X'$ such that $|X'_i \cap S'| \leq 1$ for all $i \in \{1, \ldots, n\}$. Then,

$$\begin{aligned} f'(S') = f(h(S')) &\leq f(OPT) \qquad (10) \\ &= f(h(O)) \\ &= f'(O). \qquad (11) \end{aligned}$$

where (10) holds since $|h(S') \cap X_i| \leq 1$ for $i \in \{1, \ldots, n\}$ by construction and since OPT is the optimal solution satisfying the partition matroid constraint. Since $f'(O)$ is greater than or equal to the value of any subset $S' \subseteq X'$ that satisfies the partition matroid constraint, O is the optimal solution for f', proving the claim.

Proof of (iii): We next show that G is a worst-case greedy solution of f'. By construction, we have that $h(G) = S_\mathcal{G}$. Let the information available to agent i when executing the greedy strategy on f' be denoted by $X'_{\text{in}}(i, \mathcal{G}) = \{g_j | j \in \mathcal{N}(i, \mathcal{G})\}$. By construction,

$$h(X'_{\text{in}}(i, \mathcal{G})) = X_{\text{in}}(i, \mathcal{G}),$$

for all $i \in \{1, \ldots, n\}$, where $X_{\text{in}}(i, \mathcal{G}) = \{x_j | j \in \mathcal{N}(i, \mathcal{G})\}$ is the information available to agent i when executing the greedy strategy on f. Now since $f'(S') = f(h(S'))$ for all $S' \subseteq X'$, we also have that $f'(g_i|X'_{\text{in}}(i,\mathcal{G})) = f(x_i|X_{\text{in}}(i,\mathcal{G}))$ for all $i \in \{1,\dots,n\}$.

We now need to verify that

$$g_i \in \arg\max_{x' \in X'_i} f'(x|X'_{\text{in}}(i,\mathcal{G})). \tag{12}$$

Suppose, by way of contradiction that g_i is not a maximizer of (12), i.e., there exist $x'_i \in X'_i$ such that

$$f'(x'_i|X'_{\text{in}}(i,\mathcal{G})) > f'(g_i|X'_{\text{in}}(i,\mathcal{G})).$$

This would imply that $f(h(x'_i)|X_{\text{in}}(i,\mathcal{G})) > f(x_i|X_{\text{in}}(i,\mathcal{G}))$ and since $h(x'_i) \in X_i$, we conclude that x_i is not a valid choice of the greedy algorithm on f, which is a contradiction. Therefore, G is a greedy solution for f'. Moreover, all greedy solutions for f' have the same objective value, and thus G is a worst-case solution. To see this, take any greedy solution $G' = \{g'_1,\dots,g'_n\}$ for f'. Notice that $g'_i = g_i$ if $p(g_i) \neq p(o_i)$ and $g'_i \in \{g_i, o_i\}$ if $p(g_i) = p(o_i)$. Thus, we see that $h(G') = S_{\mathcal{G}}$ and thus $f'(G') = f'(G)$.

Proof of (iv) and (v): Finally, we have that $f'(O) = f(OPT)$ and $f'(G) = f(S_{\mathcal{G}})$, concluding the proof. ■

Using this lemma we prove our main result.

Proof of Theorem V.1. Consider an arbitrary normalized monotone and submodular function $f : 2^X \to \mathbb{R}$ with action sets $X_1,\dots,X_n$ and $X = \cup_{i=1}^n X_i$ and let $\mathcal{G}$ be the information graph. Let $S_{\mathcal{G}}$ be a worst-case greedy solution i.e., $S_{\mathcal{G}} \in \arg\min_{S \subseteq \mathbb{S}_{\mathcal{G}}} f(S)$ and let OPT be an optimal solution.

Let $X'_i = \{g_i, o_i\}$ for each $i \in \{1,\dots,n\}$ and $X' = \cup_{i=1}^n X'_i$. Moreover, let $O = \{o_1,\dots,o_n\}$ and $G = \{g_1,\dots,g_n\}$. We now show that there exists a vector v in the feasible region of linear program (8) with $A = O, B = G$, $\mu = 1$, and information graph $\mathcal{G}$, such that the corresponding function f_v has the same competitive ratio as f.

Given f we apply Lemma V.2, to produce a function $f' : 2^{X'} \to \mathbb{R}$, with with worst-case greedy solution G given $\mathcal{G}$ and optimal solution O where $G \cap O = \emptyset$. We also know that $f(S_{\mathcal{G}}) = f'(G)$ and $f(OPT) = f'(O)$.

Note that $\mu = 1$ in the linear program, and so all feasible vectors v have corresponding functions f_v with $f_v(G) = 1$. Since $f'(G)$ is not necessarily 1, we define the scaled version of f' by dividing all function values by $f'(G)$, i.e., $f'_{\text{scaled}}(S') = \frac{1}{f'(G)} f'(S')$ for all $S' \subseteq X'$.

Let $v \in \mathbb{R}^{2^{2n}}$ with $v_{S'} = f'_{\text{scaled}}(S')$ for all $S' \subseteq X'$, then v is a vector in the feasible region of the linear program because, the corresponding function $f_v = \frac{1}{f'(G)} f'$ is normalized, monotone and submodular with greedily selected solution B and $f_v(B) = 1$.

Using this fact we next show that the optimal solution of the linear program produces a function with the worst-case competitive ratio: Let $v^* \in \mathbb{R}^{2^{2n}}$ be the optimal solution to the linear program given $A, B, \mathcal{G}$ and $\mu = 1$ and let f_{v^*} be its corresponding function. We know that

$$\gamma(f_{v^*}, X', \mathcal{G}) = \frac{f_{v^*}(B)}{f_{v^*}(A)}.$$

We also have that,

$$\frac{f(S_{\mathcal{G}})}{f(OPT)} = \frac{f'(G)}{f'(O)} = \frac{f'_{\text{scaled}}(G)}{f'_{\text{scaled}}(O)} = \frac{1}{f_v(A)} \tag{13}$$

$$\geq \frac{1}{f_{v^*}(A)} \tag{14}$$

$$= \gamma(f_{v^*}, X', \mathcal{G}), \tag{15}$$

where (13) and (14) holds since v is in the feasible region of the program, and v^* is the solution to the linear program, which means $v_A \leq v^*_A$ and $f_v(A) \leq f_{v^*}(A)$. Equation (15), since $f_{v^*}(B) = 1$, $\frac{1}{f_{v^*}(A)} = \gamma(f_{v^*}, X', \mathcal{G})$. As a result, we conclude that $\frac{f(S_{\mathcal{G}})}{f(OPT)} \geq \gamma(f_{v^*}, X', \mathcal{G})$ Since the inequality holds for all normalized monotone, and submodular functions defined over an arbitrary X, the result follows. ■

This result enables the use of linear programming to produce worst-case functions for distributed submodular maximization problems. For problems with a small number of agents we can directly solve the LP to produce worst-case functions. Additionally, we believe the LP formulation is of independent interest as it provides a complete characterization of worst-case functions. The following section highlights an instance of each of these applications.

VI. Worst-Case Studies

In this section we apply the results from Section V to provide two insights into the performance of greedy algorithms in distributed submodular maximization.

A. Size of Worst-Case Functions

An immediate consequence of Theorem V.1 on the size of the domain of worst-case functions is stated next.

Corollary VI.1. *Given n agents and an information graph $\mathcal{G}$, the worst-case competitive ratio $\gamma(\mathcal{G})$ for Problem* (5) *is achieved with a submodular function f defined over a base set X of just $2n$ elements, where each agent's action set has size two.*

The definition of the competitive ratio in (6) (originally proposed in [18]) uses an $\inf$ for the reason that the base set could potentially become very large. However, our result establishes that this is not the case and a worst-case function for n agents can be produced on a base set of only $2n$ elements, and where each agent has just two options in its action set. As a result, the competitive ratio for the partition matroid case can be defined with a $\min$ instead of an $\inf$.

B. Benefits of Information in Distributed Submodular Maximization

For a *fixed* submodular function, it is easy to construct examples where the performance of the distributed greedy algorithm degrades when an edge is added to the information graph. For instance, consider a coverage problem where each agent has a choice of discs, and the objective is to maximize the area of the union of the selected discs, as shown in Figure 1. The coverage function is normalized, monotone and submodular. In this example, agent 1 and 3 have just one element in their action set. If agent 2 knows the selection of

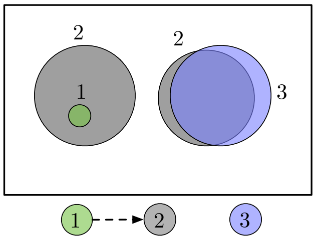

Fig. 1. Submodular function where adding an edge to information graph degrades performance. The information graph is shown at the bottom.

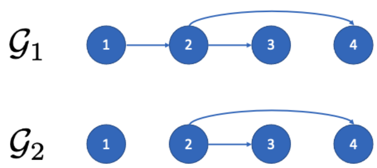

Fig. 2. Example graphs $\mathcal{G}_1$ and $\mathcal{G}_2$, where $\mathcal{G}_1$ has better competitive ratio with one less edge than $\mathcal{G}_2$.

agent 1, it will select the grey disc on the right. If it does not, it will select the grey disc on the left. The resulting area covered is smaller when agent 2 does has access to the choice of agent 1.

In what follows we present a stronger result. We show that adding an edge to the information graph can, in some cases, degrade the *worst-case performance* of the distributed greedy algorithm. One might find these results surprising as, unlike the "information never hurts" principle in probability and estimation [21], we find that information can hurt in distributed submodular maximization.

Consider the graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ defined over four agents where as shown in Figure 2. Note that $\mathcal{G}_2$ is obtained by removing a single edge from $\mathcal{G}_1$. Using our linear programming formulation, we have constructed worst-case functions for each graph[1]. The competitive ratios of $\mathcal{G}_1$ and $\mathcal{G}_2$ respectively are $\gamma(\mathcal{G}_1) = 0.250$ and $\gamma(\mathcal{G}_2) = 0.333$ (quoted to three decimal places), and thus $\gamma(\mathcal{G}_2) > \gamma(\mathcal{G}_1)$. Said differently, on graph $\mathcal{G}_1$ there is a function where the distributed greedy produces a solution achieves 0.250 of optimal, while on $\mathcal{G}_2$ there does not exist a function where the distributed greedy performs worse than 0.333 of optimal.

VII. Conclusions

We establish a framework for casting worst-case problems in submodular maximization as linear programs. We directly apply this observation to distributed submodular maximization problem, proving that for an information graph, finding the worst-case performance of the distributed strategy is equivalent to solving a linear program. We provide results capturing the structure of worst-case submodular functions that demonstrate how adding and removing edges can affected the worst-case performance.

[1]For each function, we provide the full list of all 256 function values at `https://ece.uwaterloo.ca/~sl2smith/functions_values.txt`.

References

[1] Z. Liu, A. Clark, P. Lee, L. Bushnell, D. Kirschen, and R. Poovendran, "Submodular optimization for voltage control," *IEEE Transactions on Power Systems*, vol. 33, no. 1, pp. 502–513, 2018.

[2] J. Qin, I. Yang, and R. Rajagopal, "Submodularity of storage placement optimization in power networks," *IEEE Transactions on Automatic Control*, vol. 64, no. 8, pp. 3268–3283, 2019.

[3] V. Tzoumas, M. A. Rahimian, G. J. Pappas, and A. Jadbabaie, "Minimal actuator placement with bounds on control effort," *IEEE Transactions on Control of Network Systems*, vol. 3, no. 1, pp. 67–78, 2016.

[4] V. Tzoumas, A. Jadbabaie, and G. J. Pappas, "Robust and adaptive sequential submodular optimization," *IEEE Transactions on Automatic Control*, pp. 1–1, 2020.

[5] M. Roberts, S. Shah, D. Dey, A. Truong, S. Sinha, A. Kapoor, P. Hanrahan, and N. Joshi, "Submodular trajectory optimization for aerial 3d scanning," in *IEEE Int. Conf. on Computer Vision (ICCV)*, 2017, pp. 5334–5343.

[6] J.-J. Wu and K.-S. Tseng, "Adaptive submodular inverse reinforcement learning for spatial search and map exploration," *Autonomous Robots*, vol. 46, no. 2, pp. 321–347, 2022.

[7] 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.

[8] ——, "Submodularity and greedy algorithms in sensor scheduling for linear dynamical systems," *Automatica*, vol. 61, pp. 282–288, 2015.

[9] L. F. O. Chamon, G. J. Pappas, and A. Ribeiro, "The mean square error in Kalman filtering sensor selection is approximately supermodular," in *IEEE Conference on Decision and Control (CDC)*, 2017, pp. 343–350.

[10] A. Krause, J. Leskovec, C. Guestrin, J. VanBriesen, and C. Faloutsos, "Efficient sensor placement optimization for securing large water distribution networks," *Journal of Water Resources Planning and Management*, vol. 134, no. 6, pp. 516–526, 2008.

[11] X. Sun, C. G. Cassandras, and X. Meng, "A submodularity-based approach for multi-agent optimal coverage problems," in *IEEE Conference on Decision and Control (CDC)*, 2017, pp. 4082–4087.

[12] L. Zhou, V. Tzoumas, G. J. Pappas, and P. Tokekar, "Resilient active target tracking with multiple robots," *IEEE Robotics and Automation Letters*, vol. 4, no. 1, pp. 129–136, 2019.

[13] A. Hashemi, M. Ghasemi, H. Vikalo, and U. Topcu, "Randomized greedy sensor selection: Leveraging weak submodularity," *IEEE Transactions on Automatic Control*, vol. 66, no. 1, pp. 199–212, 2021.

[14] A. Downie, B. Gharesifard, and S. L. Smith, "Submodular maximization with limited function access," 2022, arXiv:2201.00724.

[15] L. Lovász, "Submodular functions and convexity," in *Mathematical programming the state of the art*. Springer, 1983, pp. 235–257.

[16] 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.

[17] B. Gharesifard and S. L. Smith, "Distributed submodular maximization with limited information," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 4, pp. 1635–1645, 2018.

[18] D. Grimsman, M. S. Ali, J. P. Hespanha, and J. R. Marden, "The impact of information in distributed submodular maximization," *IEEE Transactions on Control of Network Systems*, vol. 6, no. 4, pp. 1334–1343, 2019.

[19] D. Grimsman, M. R. Kirchner, J. P. Hespanha, and J. R. Marden, "The impact of message passing in agent-based submodular maximization," in *IEEE Conference on Decision and Control (CDC)*, 2020, pp. 530–535.

[20] M. Corah and N. Michael, "Distributed submodular maximization on partition matroids for planning on large sensor networks," in *IEEE Conference on Decision and Control (CDC)*, 2018, pp. 6792–6799.

[21] T. M. Cover, J. A. Thomas *et al.*, "Entropy, relative entropy and mutual information," *Elements of information theory*, vol. 2, no. 1, pp. 12–13, 1991.