The Impact of Information in Distributed Submodular Maximization

David Grimsman [ORCID], Mohd. Shabbir Ali [ORCID], João P. Hespanha [ORCID], and Jason R. Marden [ORCID]

Abstract—The maximization of submodular functions is an NP-Hard problem for certain subclasses of functions, for which a simple greedy algorithm has been shown to guarantee a solution whose quality is within 1/2 of the optimal. When this algorithm is implemented in a distributed way, agents sequentially make decisions based on the decisions of all previous agents. This work explores how limited access to the decisions of previous agents affects the quality of the solution of the greedy algorithm. Specifically, we provide tight upper and lower bounds on how well the algorithm performs, as a function of the information available to each agent. Intuitively, the results show that performance roughly degrades proportionally to the size of the largest group of agents that make decisions independently. Additionally, we consider the case where a system designer is given a set of agents and a global limit on the amount of information that can be accessed. Our results show that the best designs partition the agents into equally sized sets and allow agents to access the decisions of all previous agents within the same set.

Index Terms—Distributed algorithms, graph theory, greedy algorithms, multiagent systems, optimization.

I. INTRODUCTION

THE optimization of submodular functions is a well-studied topic due to its application in many common engineering problems. Examples include information gathering [2], maximizing influence in social networks [3]; image segmentation in image processing [4]; multiple object detection in computer vision [5]; document summarization [6]; path planning of multiple robots [7]; sensor placement [8], [9]; and resource allocation in multiagent systems [9]. The key thread in these problems is that each exhibits some form of a "diminishing returns" property, for example, adding more sensors to a sensor placement problem improves performance, but every additional sensor marginally contributes less to the overall performance as the number of sensors increases. Any problem exhibiting such behavior can likely be formulated as a submodular optimization problem.

While polynomial algorithms exist to solve submodular minimization problems, [10]–[12], maximization has been shown to be NP-Hard for important subclasses of submodular functions [13]. Thus, a tremendous effort has been placed on developing fast algorithms that approximate the solution to the submodular maximization problem [14]–[20]. A resounding message from this extensive research is that very simple algorithms can provide strong guarantees on the quality of the approximation.

The seminal work in [15] demonstrates that a greedy algorithm provides a solution that is within $1/2$ of the quality of the optimal solution. In fact, more sophisticated algorithms can often be derived for certain classes of submodular maximization problems that push these guarantees from $1/2$ to $1 - 1/e$ [14], [21], [22]. Progress beyond this level of suboptimality is not possible in general, because it was also shown that no polynomial-time algorithm can achieve a higher guarantee than $(1 - 1/e)$, unless $P = NP$ [23].

One appealing trait of the greedy algorithm is that it can be implemented in a distributed way while still maintaining the $1/2$ performance guarantee. In the distributed greedy algorithm, each agent in the set sequentially selects its choice by greedily optimizing the global objective function conditioned on the decisions of the previous agents. However, this requires agents to have full access to the decisions of all previous agents. In addition, each agent must have access to the value of the global objective function for previous agents' decisions, plus any choice in its own decision set. In many cases, these informational demands may be costly or infeasible.

Research has therefore begun to explore how limited information can impact the performance of the greedy algorithm on distributed submodular maximization problems. For example, Gairing [9] focused on the submodular resource allocation problem, modeled as a game played among agents. The resulting Nash equilibria have the familiar $1/2$ performance guarantee; however, it is shown that when information is limited to be local instead of global, the performance guarantee degrades to $1/n$, where n is the number of agents. The work in [24] formulates the problem of selecting representative data points from a large corpus as a submodular maximization problem. In order to perform the optimization in a distributed way, agents are partitioned into sets, where the full greedy algorithm is performed among agents within a set, while no information is transferred between sets.

In this setting, the paper shows that the algorithm performance is worse than $1/2$, even when a preprocessing algorithm is used to intelligently assign decision sets to each agent. Other work in [20] discusses the role of information in the task assignment problem. It is shown that the distributed greedy algorithm can be implemented asynchronously, with convergence in a finite number of steps. Additionally, when agent action sets are based on spatial proximity, agents need only consider local information to achieve the $1/2$ bound. Finally, the work in [25] studied the performance of the distributed greedy algorithm when an agent can only observe a local subset of its predecessors. It is shown that localizing information, particularly when agents are partitioned from each other, leads to a degradation in performance. For instance, in the case where agents are partitioned into sets, performing the full greedy algorithm within the set and obtaining no information outside the set, the performance degrades proportionally to the number of sets in the partition.

This paper more closely relates to the work done in [25] in evaluating informational constraints. We leverage a similar model and seek to find how limiting, which decisions an agent can access, impacts the overall performance of the distributed greedy algorithm. We also consider the scenario where a system designer is given a set of agents and a global limit on the amount of information that can be accessed, and seek to find the best policies to ensure the highest performance possible.

More specifically, the contributions of this paper are the following results.

1) Theorem 1 gives lower and upper bounds on worst-case performance of the greedy algorithm on any submodular function for any given set of constraints. The bounds show that worst-case performance (roughly) degrades proportionally to the size of the largest group of agents that make decisions independently.

2) Theorem 2 shows the best performance of the greedy algorithm that a system designer can achieve with a fixed number of agents and constraints. The results show that when information is costly, the best system design is to partition the agents into equal sets, and have agents in each set execute the full greedy algorithm.

The remainder of this paper is dedicated to proving and discussing these two theorems.

II. MODEL

This paper focuses on a distributed algorithm for solving submodular maximization. To that end, let S be a set of elements and $f : 2^S \to \mathbb{R}_{\geq 0}$ have the following properties.

1) *Normalized*: $f(\emptyset) = 0$.
2) *Monotonic*: For $A \subseteq B \subseteq S$, $f(A) \leq f(B)$.
3) *Submodular*: For $A \subseteq B \subset S$ and $x \in S \setminus B$, the following holds:

$$f(A \cup \{x\}) - f(A) \geq f(B \cup \{x\}) - f(B). \quad (1)$$

For simplicity, we will refer to a function with all three above properties merely as submodular.

In this paper, we focus on distributed approaches to submodular optimization where there are a set of decision-making agents $N = \{1, 2, \ldots, n\}$, and each agent i is associated with an *ac-*

tion set $X_i \subseteq 2^S$. Notationally, we define $X = X_1 \times \cdots \times X_n$ as the family of action sets, an *action* for agent i as $x_i \in X_i$, and an *action profile* as $x \in X$. We also overload the notation of f to allow multiple actions as inputs: $f(x_i, x_j) := f(x_i \cup x_j)$, $f(x) := f(\bigcup_i x_i)$ and $f(x_{a:b}) := f\left(\bigcup_{a \leq i \leq b} x_i\right)$ for $a \leq b$. The submodular maximization problem addressed in this work is to find

$$x^{\text{opt}} \in \max_{x \in X} f(x). \quad (2)$$

As shown in [25], this type of constraint where we choose from a family of subsets X_i is also referred to in the literature as a partition matroid constraint.

We next present two relevant problems that can be modeled accordingly. This serves to give a scope and relevance to the model, as well as provide an example that will be leveraged throughout the rest of the paper.

Example 1 (Vehicles target assignment problem [26]): Consider the classic vehicles target assignment problem where there are a collection of targets $\mathcal{T}$ and each target $t \in \mathcal{T}$ has an associated value $v_t \geq 0$. Further, there exists a collection of n agents, and each agent i is associated with a success probability $p_i \in [0, 1]$ and a set of possible assignments $X_i \subseteq 2^{\mathcal{T}}$. The agents make decisions to reach a feasible allocation of agents to targets $x = (x_1, \ldots, x_n) \in X_1 \times \cdots \times X_n$ that optimizes a system-level performance metric of the form

$$W(x) = \sum_{t \in \cup_i x_i} v_t \left(1 - \prod_{i : t \in x_i} (1 - p_i)\right). \quad (3)$$

Note that the objective function given in (3) is submodular, as W can be expressed as a function of the form $W : 2^S \to \mathbb{R}_{\geq 0}$ for an appropriate choice of the domain set S, that is, $S = N \times 2^{\mathcal{T}}$ and the action sets can be expressed as disjoint sets in S.

Example 2 (Weighted set cover problem [27]): Consider the subset of the vehicles target assignment problem where $p_i = 1$ for all i. Then, the objective function takes on the form

$$W(x) = \sum_{t \in \cup_i x_i} v_t. \quad (4)$$

Note that (4) can now be expressed by a submodular function with domain $S = \mathcal{T}$. An instance of such a problem is shown in Fig. 1.

We will henceforth focus on submodular functions of the form $f : 2^S \to \mathbb{R}_{\geq 0}$, where $X_i \subseteq 2^S$, without explicitly highlighting the structure of S and the action sets $X_1, \ldots, X_n$. Furthermore, we will rely on the weighted set cover problem in several of the forthcoming proofs.

III. GREEDY ALGORITHM

One of the most well-studied algorithms to solve the submodular maximization problem is the greedy algorithm. This algorithm requires agents to make decisions sequentially,[1] so

---

[1]Although we state that the agents must choose sequentially, the real restriction is that the flow of information in the system is acyclic. Accordingly, if the agents select their decisions with regards to another process, e.g., a synchronous best reply process, they will still arrive at the same solution. Once agent 1 has chosen, it will not make a different choice regardless of order, and agent 2 will not switch after agent 1 has decided, etc. See [25] for more details).

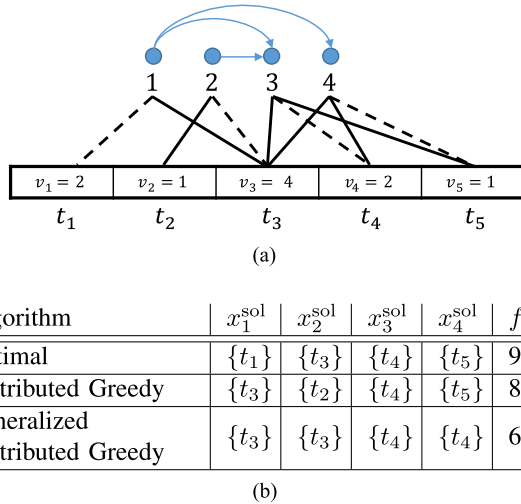| Algorithm	x_1^{sol}	x_2^{sol}	x_3^{sol}	x_4^{sol}	$f(x^{\mathrm{sol}})$
Optimal	$\{t_1\}$	$\{t_3\}$	$\{t_4\}$	$\{t_5\}$	9
Distributed Greedy	$\{t_3\}$	$\{t_2\}$	$\{t_4\}$	$\{t_5\}$	8
Generalized Distributed Greedy	$\{t_3\}$	$\{t_3\}$	$\{t_4\}$	$\{t_4\}$	6

(b)

Fig. 1.   Instance of the weighted set cover problem and the performance of the greedy algorithm in solving it. (a) Setup of a weighted set cover problem. The targets are $\mathcal{T} = \{t_1, \dots, t_5\}$, each represented by a box, and each with a corresponding value. The available choices to each agent are represented by the black lines (both dotted and solid)—for instance $X_1 = \{\{t_1\}, \{t_3\}\}$, $X_2 = \{\{t_2\}, \{t_3\}\}$, etc. The dashed lines represent an optimal set of choices. The goal for the agents is to maximize $W(x)$ in (4). Using the generalized distributed algorithm [i.e., agents choose according to (7)], agent 1 chooses t_3, since $v_3 > v_1$. Then, agent 2, who (according to the graph) does not know that agent 1 has chosen t_3, also chooses t_3, since $v_3 > v_2$. Agent 3 observes that agents 1 and 2 have both chosen t_3, so it chooses t_4, since $v_4 > v_5$. Finally, agent 4, observing that agent 1 has chosen t_3 (but not that agent 3 has chosen t_4), chooses t_4, since $v_4 > v_5$. These results are summarized in the table below. (b) For the weighted set cover problem outlined above, this table shows the agents' decisions in an optimal case, the case where the distributed greedy algorithm is used [agents choose according to (5)] and the case where the generalized distributed algorithm is used (agents choose according to (7), constrained to the graph shown above). The difference between the distributed greedy algorithm and the generalized version can be seen in the choices of agents 2 and 4. Agent 2 chooses t_3 when it can observe that t_4 has already been chosen by agent 1, otherwise it chooses t_4. Likewise, agent 4 chooses t_5 only when it knows that t_4 has already been selected. Therefore, as the informational constraints grow, the solution quality decreases. As a note, in this case, we see that $\gamma(f, X, G) = 6/9$.

without loss of generality, we impose an ordering on the agents according to their labels, that is, agent 1 chooses first, agent 2 chooses second, etc. Agent i makes its choice $x_i^{\mathrm{sol}} \in X_i$ based on the following rule:

$$x_i^{\mathrm{sol}} \in \arg\max_{x_i \in X_i} f\left(x_i, x_{1:i-1}^{\mathrm{sol}}\right). \tag{5}$$

In words, each agent i selects the action that would optimize the objective function f given knowledge of the action choices of the previous agents and the global objective function f.

It is well-known that for any submodular f, any set of action spaces X, and any order of agents, the quality of the resulting solution x^{sol} derived from the distributed greedy algorithm compared to the optimal solution x^{opt} satisfies

$$\frac{f(x^{\mathrm{sol}})}{f(x^{\mathrm{opt}})} \geq \frac{1}{2} \tag{6}$$

where $x^{\mathrm{sol}} = (x_1^{\mathrm{sol}}, \dots, x_n^{\mathrm{sol}})$. In other words, the quality of the solution is within $1/2$ that of the optimal [14]. For special classes of submodular functions, and additional constraints placed on

$X_1, \dots, X_n$ (for instance if $X_1 = \dots = X_n$), [14] also shows that the solution to the greedy lies within $1 - 1/e \approx 63\%$ of the optimal.

In (5), agent i must have access to the decisions of all previous agents. However, there are many applications where this level of informational demand may be impractical. A more generalized version of the distributed greedy algorithm is proposed in [25], where each agent i makes its choice using the following rule:

$$x_i^{\mathrm{sol}} \in \arg\max_{x_i \in X_i} f\left(x_i, x_{\mathcal{N}_i}^{\mathrm{sol}}\right) \tag{7}$$

where $\mathcal{N}_i \subseteq \{1, \dots, i-1\}$, $\mathcal{N}_1 = \emptyset$, and $x_{\mathcal{N}_i} = \cup_{j \in \mathcal{N}_i} x_j$. The sets $\mathcal{N}_1, \dots, \mathcal{N}_n$ characterize the informational constraints of the agent in the sense that $\mathcal{N}_i$ is the set of agents whose choices agent i can access when making its own action decision. The central topic of discussion here is how the structure of $\mathcal{N}_1, \dots, \mathcal{N}_n$ impacts the performance guarantees associated with this generalized greedy algorithm. Note that (7) is simply one decision rule that could be used by the agents. Analysis of whether this rule is optimal among all possible decision rules given the local information is the topic of ongoing research.

It is helpful to model the informational constraints as a graph $G = (V, E)$, where V is a set of nodes and $E \subseteq V \times V$ is a set of directed edges between nodes. In this scenario, each node is an agent (and thus we use the terms interchangeably) and each edge (j, i) implies that $j \in \mathcal{N}_i$, i.e., $\mathcal{N}_i$ is the set of in-neighbors for vertex i. Since there is an imposed ordering on the vertices, and the agents choose sequentially, the set $\mathcal{G} := \{G = (V, E) : (i, j) \in E \Rightarrow i < j\}$ is the set of admissible graphs that correspond to a set of informational constraints.

The solution of the algorithm defined by (7) is denoted $x^{\mathrm{sol}}(f, X, G)$ and the optimal decisions as $x^{\mathrm{opt}}(f, X)$ to explicitly highlight the dependence of (7) on the graph G. We define the efficiency guarantees associated with this algorithm as

$$\gamma(f, X, G) := \frac{f(x^{\mathrm{sol}}(f, X, G))}{f(x^{\mathrm{opt}}(f, X))}. \tag{8}$$

Note that $x^{\mathrm{sol}}(f, X, G)$ could in fact be a set when (7) is not unique, so we write $f(x^{\mathrm{sol}}(f, X, G))$ with the understanding that f is evaluated at the worst possible candidate solution, i.e., $\min_{x \in x^{\mathrm{sol}}(f, X, G)} f(x)$.

One goal of this paper is to characterize the efficiency guarantees associated with this more generalized version of the distributed greedy algorithm for any submodular function and action sets X. To that end, we define

$$\gamma(G) := \inf_{f, X} \gamma(f, X, G). \tag{9}$$

In words, $\gamma(G)$ is the worst-case efficiency for any f and family of sets $X_1, \dots, X_n$ as defined above, given the informational constraints among the agents represented by G.

IV. EFFICIENCY BOUNDS

In this section, we present lower and upper bounds for the worst-case efficiency $\gamma(G)$ based on the structure of the graph $G \in \mathcal{G}$. We begin with some preliminaries from graph theory, and then present the bounds.

A. Preliminaries

For all definitions in this section, we assume that $G = (V, E)$ is a general directed graph. We begin with cliques: a *clique* is a set of nodes $C \subseteq V$ such that for every $i, j \in C$, either $(i, j) \in E$ or $(j, i) \in E$. The *clique number* $\omega(G)$ is the number of nodes in the largest clique in G. We denote by $K(G)$ the set of all cliques in G. A *clique cover* is a partition on V such that the nodes in each set of the partition form a clique. The *clique cover number* $k(G)$ is the minimum number of sets needed to form a clique cover of G. For an example, see Fig. 2(a).

Another important notion in graph theory is that of independence. An *independent set* $J \subseteq V$ is a set of vertices such that $v_1, v_2 \in J$ implies $(v_1, v_2), (v_2, v_1) \notin E$. A *maximum independent set* is an independent set of G such that no other independent set has more vertices. The *independence number* $\alpha(G)$ is the number of nodes in the largest independent set in G. For an example, see Fig. 2(a).

The work in [28] equivalently characterizes the independence number as the solution to an integer linear program.[2] Let $W \in \mathbb{R}^{|K(G)| \times n}$ be the binary matrix whose rows are indicator vectors for the cliques in G. In other words, $W_{ij} = 1$ if node j belongs to clique i in G, and 0 otherwise. Note that W also includes cliques of size 1 (the individual nodes). Then, $\alpha(G)$ is given by

$$\max_z \quad z^T \mathbf{1}$$
$$\text{subject to} \quad Wz \leq \mathbf{1}$$
$$z \in \mathbb{Z}^n \geq \mathbf{0}. \tag{10}$$

It is similarly shown that $k(G)$ is characterized by the dual to this problem, implying that $\alpha(G) \leq k(G)$. As an example, for the graph in Fig. 2(a)

$$
\begin{array}{c}
\text{Node 1 \ Node 2 \ Node 3 \ Node 4} \\
W = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
1 & 1 & 0 & 0 \\
1 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 \\
0 & 1 & 1 & 0 \\
0 & 1 & 0 & 1 \\
1 & 1 & 1 & 0 \\
1 & 1 & 0 & 1
\end{bmatrix}
\begin{array}{l}
\{1\} \\
\{2\} \\
\{3\} \\
\{4\} \\
\{1,2\} \\
\{1,3\} \\
\{1,4\} \\
\{2,3\} \\
\{2,4\} \\
\{1,2,3\} \\
\{1,2,4\}.
\end{array}
\end{array} \tag{11}
$$

Using this W in (10), it is straightforward to show that the optimal solution is $z = [0, 0, 1, 1]^T$, i.e., $\alpha(G) = 2$, and the maximum independent set is $\{3, 4\}$.

Note by definition that $\alpha(G)$ and $k(G)$ are always positive integers. However, in many applications, it is helpful to consider a real-valued relaxation on these notions: this is the motivation for fractional graph theory [28]. Here, we leverage the *fractional independence number* $\alpha^*(G)$, which we define as the real-valued relaxation to (10):[3]

$$\alpha^*(G) := \max_z \quad z^T \mathbf{1}$$
$$\text{subject to} \quad Wz \leq \mathbf{1}$$
$$z \geq \mathbf{0}. \tag{12}$$

Likewise, $k^*(G)$, the *fractional clique cover number* of G, can be defined by its dual

$$k^*(G) := \min_y \quad y^T \mathbf{1}$$
$$\text{subject to} \quad W^T y \geq \mathbf{1}$$
$$y \geq \mathbf{0}. \tag{13}$$

In accordance with the Strong Duality of Linear Programming, it follows that

$$\alpha(G) \leq \alpha^*(G) = k^*(G) \leq k(G). \tag{14}$$

An example of a graph where the independence number differs from the fractional independence number is found in Fig. 2(b).

B. Result

We now present results regarding the quality of the solution provided by the generalized distributed greedy algorithm subject to the informational constraints represented by a graph G. We show that the performance degrades proportionally to the fractional independence number of G.[4]

Theorem 1: For any graph $G \in \mathcal{G}$

$$\frac{1}{\alpha^*(G)} \geq \gamma(G) \geq \frac{1}{\alpha^*(G) + 1}. \tag{15}$$

The upper bound shows that it is impossible to construct a graph G such that the greedy algorithm's performance is better than $1/\alpha^*(G)$ for all possible f and X. Likewise, the lower bound means that no f and X can result in a performance lower than $1/(\alpha^*(G) + 1)$.

The formal proof for this theorem is given in Section IV-D, but here we give a brief outline of the argument. For the upper bound, we develop a canonical f and X, which are dependent on the cliques in G. Then, we show that for this example, $\gamma(f, X, G)$ is the inverse of the solution to (12), i.e., $\alpha^*(G)$. The lower bound is found by leveraging the properties of submodularity and monotonicity, showing that the highest lower bound requires solving (13). Then, using (14), we show that Theorem 1 holds.

C. Examples

Theorem 1 shows lower and upper bounds on $\gamma(G)$, but we have not shown whether either of these bounds is tight. There

---

[3]Another definition of fractional independence exists in the literature (see [29]), which was created to preserve certain properties of graph independence (such as nested maximality), but has not been shown to preserve $\alpha^*(G) = \omega^*(\bar{G})$, where $\bar{G}$ is the complement graph of G and $\omega^*(\bar{G})$ is the fractional clique number of G.

[4]In comparison to the bounds shown in [25], the bounds shown in our work are tighter in all cases. In fact, except in certain corner cases (for example, both bounds are the same on a full clique), our results are strictly tighter.

---

[2]It is actually the chromatic number and clique number that are defined this way in [28]. However, using graph complementarity, it is an easy extension to show that the solution to the linear program in (10) yields a maximum independent set.
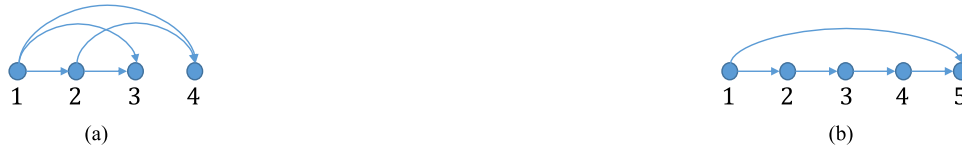
Fig. 2.   Two example graphs showcasing the graph properties defined in Section IV-A. These graphs will be referred to throughout the paper to illustrate the tightness of bounds in Theorem 1 and to illustrate the Sibling Property (see Section V-C). (a) In this graph, there are four cliques of size 1 (one for each node), five cliques of size 2 (one representing each edge), and two cliques of size 3 (the sets $\{1, 2, 3\}$ and $\{1, 2, 4\}$). Thus, $\omega(G) = 3$. A minimum clique cover is $\{1, 3\}, \{2, 4\}$, so $k(G) = 2$. The maximum independent set is $\{3, 4\}$, thus $\alpha(G) = 2$. Since $\alpha(G) = k(G) = 2$, we know that $\alpha^*(G) = k^*(G) = 2$. Appendix A shows that $\gamma(G) = 1/2$, making it a graph that meets the upper bound for Theorem 1 (see Section IV-C). Finally, it is also an example of a graph without the Sibling Property (see Section V-C), since no such w exists from Definition 1. (b) Graph where $\alpha(G) = 2$, $k(G) = 3$, $\alpha^*(G) = k^*(G) = 2.5$, and $z = [1/2, 1/2, 1/2, 1/2, 1/2]^T$ maximizes (12). As a note, this is the graph with the fewest number of nodes and edges such that $\alpha(G) \neq k(G)$. This is also a graph with the Sibling Property (see Section V-C), since for maximum independent set $\{2, 4\}$, $2 \in \mathcal{N}_3$, so $w = 3$ from Definition 1.
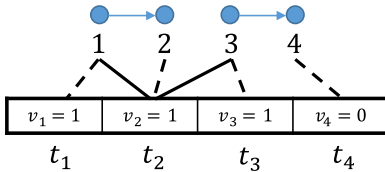

Fig. 3.   Example of a graph G where $\gamma(G) = 1/(\alpha^*(G) + 1)$, and an instance of a weighted set cover problem using the same notation as in Fig. 1. Here, $\alpha(G) = \alpha^*(G) = 2$, and we can see that $f(x^{\mathrm{opt}}) = 3$. The worst-case results from the generalized distributed greedy algorithm occur when $x_1^{\mathrm{sol}} = x_2^{\mathrm{sol}} = x_3^{\mathrm{sol}} = \{t_2\}$, and therefore $f(x^{\mathrm{sol}}) = 1$. This means $\gamma(f, X, G) = 1/(\alpha^*(G) + 1) = 1/3$, so the lower bound in Theorem 1 is tight for this graph.

exist graphs for which f and X can be chosen to meet the lower bound, and there also exist graphs whose lower bound can be proven to meet the upper bound. In this section, we provide an example of each.

Example 3: The weighted set coverage problem presented in Fig. 3 is an example showing that the lower bound from Theorem 1 is tight. For this graph G, $\alpha(G) = \alpha^*(G) = 2$. As shown, $\gamma(f, X, G) = 1/(\alpha(G) + 1)$. Since $\gamma(G) \leq \gamma(f, X, G)$ by definition, it follows that $\gamma(G) = 1/(\alpha^*(G) + 1)$.

Example 4: The graph G in Fig. 2(a) is an example where the upper bound from Theorem 1 is tight. Here $\alpha^*(G) = 2$, and it is shown in Appendix A for this graph that no f and X can be constructed to give a worse efficiency than $1/2$.

D. Proof for Theorem 1

Before starting the proof, we introduce some notation. Let $A \subseteq N$ and $x_A = \bigcup_{j \in A} x_j$. Then, we define

$$\Delta(x_i|x_A) := f(x_i, x_A) - f(x_A) \qquad (16)$$

to be the marginal contribution of the choice of agent i given the choices of the agents in set A. One property to note about Δ is the following holds when G is a clique

$$\sum_i \Delta(x_i, x_{1:i-1}) = f(x). \qquad (17)$$

We now begin the theorem proof. We first show the upper bound by constructing a canonical example f and X, which can be applied to any G, and show that $\gamma(f, X, G) = 1/\alpha^*(G)$. Then, we prove the lower bound leveraging the properties of submodularity and monotonicity.

Consider a base set of elements $S = \{u_1, \ldots, u_n, v_1, \ldots, v_n\}$ and let $X_i = \{\{u_i\}, \{v_i\}\}$ for all i. We endeavor to design submodular f so that for all i, $x_i^{\mathrm{opt}} = \{v_i\}$ and in the worst case $x_i^{\mathrm{sol}} = \{u_i\}$. Let $f : 2^S \to \mathbb{R}_{\geq 0}$ have the following restrictions.

1) $f(\{u_i\}) = f(\{v_i\}) \geq 0$.
2) $f(\{u_1, \ldots, u_n\}) \leq 1$.
3) $\Delta(\{u_i\}| \cup_{j \in A} \{u_j\}) = f(\{u_i\})$ for any $A \subseteq \mathcal{N}_i$.
4) $\Delta(\{v_i\}|B) = f(\{v_i\})$ for all $i \in N$ and $B \subseteq S \setminus \{v_i\}$.

First note that Restrictions 2 and 3 hold if and only if for all $c \in K(G)$

$$f(\cup_{i \in c} \{u_i\}) = \sum_{i \in c} f(\{u_i\}) \leq 1. \qquad (18)$$

If one sets values for all $f(\{u_i\})$, which meet this requirement, a full function on 2^S can be created with a simple extension. Let $A \subseteq S$, then define

$$f(A) = \max\left\{1, \sum_{u_i \in A} f(\{u_i\})\right\} + \sum_{v_i \in A} f(\{v_i\}). \qquad (19)$$

We claim that the function f created in this way is normalized, monotone, and submodular. For $A \subseteq B \subset S$, we know by Restriction 4 that for $v_i \notin B$, $f(A, \{v_i\}) - f(A) = f(B, \{v_i\}) - f(B) = f(\{v_i\})$. For $u_i \notin B$, it should be clear from (19) that $f(A, \{u_i\}) - f(A) \geq f(B, \{u_i\}) - f(B)$. We also see that $f(B) \geq f(A)$ (monotone) and $f(\emptyset) = 0$ (normalized).

Restrictions 1, 3, and 4 imply that, since agents are choosing according to (7), every agent is choosing between equally desirable choices. However, it should be clear from Restriction 4 that $x_i^{\mathrm{opt}} = \{v_i\}$, and in the worst case $x_i^{\mathrm{sol}} = \{u_i\}$. Therefore

$$\gamma(f, X, G) = \frac{f(\{u_1\}, \ldots, \{u_n\})}{f(\{v_1\}, \ldots, \{v_n\})} \leq \frac{1}{\sum_i f(\{u_i\})} \qquad (20)$$

which is true by Restrictions 1, 2, and 4. This shows how the values $f(\{u_1\}), \ldots, f(\{u_n\})$ directly determine the upper bound on efficiency of the algorithm for this scenario.

Since by definition $\gamma(f, X, G) \geq \gamma(G)$, one can find the lowest such upper bound on $\gamma(G)$ by setting $f(\{u_i\})$ so as to maximize $\sum_i f(\{u_i\})$ [from (20)] subject to the constraint in (18). This is precisely the linear program in (12), whose value is $\alpha^*(G)$. Therefore, $1/\alpha^*(G)$ is an upper bound on $\gamma(G)$.

For the lower bound, let $x_i = x_i^{\mathrm{sol}}$ and $x = x^{\mathrm{sol}}$ for ease of notation. Then, consider the following:

$$f(x^{\mathrm{opt}}) \leq f(x, x^{\mathrm{opt}}) = f(x) + \sum_i \Delta\left(x_i^{\mathrm{opt}} \middle| \bigcup_{j<i} x_j^{\mathrm{opt}} \cup x\right) \tag{21}$$

$$\leq f(x) + \sum_i \Delta(x_i^{\mathrm{opt}}|x_{\mathcal{N}_i}) \tag{22}$$

$$\leq f(x) + \sum_i \Delta(x_i|x_{\mathcal{N}_i}) \tag{23}$$

where (22) follows from submodularity and (23) follows from the decision-making rule in (7). Let $a \in K(G)$ and let $\beta_i^P = 1 - \sum_{c \in P: i \in c} y_c$ for $P \subseteq K(G), y_c \in \mathbb{R}$. Then, it follows that

$$f(x^{\mathrm{opt}}) \leq f(x) + \sum_{i \in a} \Delta(x_i|x_{\mathcal{N}_i}) + \sum_{i \notin a} \Delta(x_i|x_{\mathcal{N}_i}) \tag{24}$$

$$\leq f(x) + y_a \sum_{i \in a} \Delta(x_i|x_{\mathcal{N}_i \cap a}) \tag{25}$$

$$+ (1-y_a) \sum_{i \in a} \Delta(x_i|x_{\mathcal{N}_i}) + \sum_{i \notin a} \Delta(x_i|x_{\mathcal{N}_i})$$

$$= f(x) + y_a f(x_a) + \sum_i \beta_i^{\{a\}} \Delta(x_i|x_{\mathcal{N}_i}) \tag{26}$$

where (25) is true by submodularity and (26) is true by (17) with some algebraic manipulation. The procedure followed in (24)–(26) can be thought of as an algorithm to put the sum in (23) into a more convenient form (for our purposes) with respect to some clique a. One could run the same procedure on the sum in (26) with respect to some clique b, to obtain

$$f(x^{\mathrm{opt}}) \leq f(x) + \sum_{c \in \{a,b\}} y_c f(x_c) + \sum_i \beta_i^{\{a,b\}} \Delta(x_i|x_{\mathcal{N}_i}). \tag{27}$$

Adding more cliques to the set $\{a, b\}$ simply requires an update on the two places that $\{a, b\}$ appears in the equation. Therefore, if the set of cliques is the full set $K(G)$, then

$$f(x^{\mathrm{opt}}) \leq f(x) + \sum_{c \in K(G)} y_c f(x_c) + \sum_i \beta_i^{K(G)} \Delta(x_i|x_{\mathcal{N}_i}). \tag{28}$$

Note that this holds for any set of y_c, but in order to remove the third term from the inequality, we impose a constraint that $\beta_i^{K(G)} \leq 0$. Alternatively stated, we require that

$$\sum_{c \in K(G): i \in c} y_c \geq 1 \quad \forall i. \tag{29}$$

Under these conditions and by monotonicity, (28) becomes $f(x^{\mathrm{opt}}) \leq f(x) + f(x) \sum_{c \in K(G)} y_c$, which implies that

$$\gamma(G) \geq \frac{1}{1 + \sum_{c \in K(G)} y_c}. \tag{30}$$

To find the highest such lower bound on $\gamma(G)$, one needs to find the minimum value for the sum in (30), subject to the constraint in (29). This is precisely the linear program defined
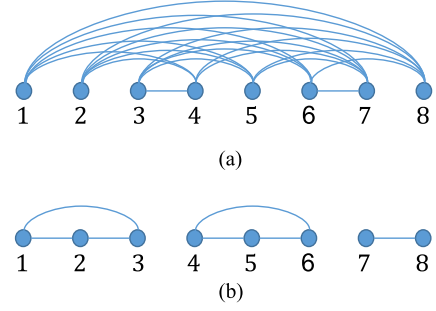

Fig. 4. Turán graph and its complement. (a) Turán graph $T(8,3)$, where $\omega(G) = 3$. No other graph with eight vertices can have more edges without also having a clique of size 4 or higher. (b) Complement Turán graph $\overline{T(8,3)}$, where $\alpha(G) = 3$. No other graph with eight vertices can have fewer edges without also having an independent set of size 4 or higher.

in (13). Therefore, it follows that $\gamma(G) \geq 1/(1 + k^*(G)) = 1/(1 + \alpha^*(G))$. $\qquad \square$

V. OPTIMAL STRUCTURES

In this section, we describe how to build a graph G that yields the highest efficiency $\gamma(G)$ subject to a constraint on the number of edges.

A. Preliminaries

We denote $\mathcal{G}_{m,n} := \{G = (V, E) \in \mathcal{G} : |V| = n, |E| \leq m\}$ and $G_{m,n}^* \in \arg\max_{G \in \mathcal{G}_{m,n}} \gamma(G)$, i.e., $G_{m,n}^*$ is a graph in $\mathcal{G}_{m,n}$ that maximizes efficiency. The complement $\bar{G} = (\bar{V}, \bar{E})$ of graph $G = (V, E)$ is such that $\bar{V} = V$ and $(i, j) \in \bar{E}$ if and only if $(i, j) \notin E$. It is straightforward to show that $\alpha(G) = \omega(\bar{G})$.

In graph theory, a *Turán graph* $T(n, r)$ is a graph with n vertices created with the following algorithm.
1) Partition the vertices into r disjoint sets $C_1, \ldots, C_r$ such that $|C_i| - |C_j| \leq 1$ for all $i, j \in \{1, \ldots, r\}$.
2) Create edges between all nodes not within the same set.

A result known as Turán's theorem states that $T(n, r)$ is an n-node graph with the highest number of edges that has clique number r or smaller [30]. Alternatively stated

$$T(n, r) \in \arg\max_{G = (V,E): \omega(G) \leq r} |E|. \tag{31}$$

The complement of a Turán graph, denoted $\overline{T(n, r)}$, is created with the same procedure as a Turán graph, except that in Step 2, edges are created among all nodes *within* the same set. An example of a Turán graph and its complement is found in Fig. 4. Thus, we can also state, similar to (31), that

$$\overline{T(n, r)} \in \arg\min_{G = (V,E): \alpha(G) \leq r} |E|. \tag{32}$$

In words $\overline{T(n, r)}$ is a graph with the fewest edges that has independence number r. It should also be clear that

$$\alpha\left(\overline{T(n, r)}\right) = \alpha^*\left(\overline{T(n, r)}\right) = k\left(\overline{T(n, r)}\right) = r. \tag{33}$$

Finally, we define the graph

$$\hat{T}(n,m) := \underset{\{\overline{T(n,r)}:|E|\leq m\}}{\arg\min} \, r \tag{34}$$

which is the complement n-node Turán graph with the lowest independence number among all graphs with the number of edges less than or equal to m.[5]

B. Result

The main result of this section regarding efficient graph structures is stated below and later proved in Section V-D.

Theorem 2: Consider two non-negative integers n and m such that $m \leq \frac{1}{2}n(n-1)$. If $m \neq \frac{1}{2}n(n-1) - 1$, then $G^*_{m,n} = \hat{T}(n,m)$. If $m = \frac{1}{2}n(n-1) - 1$, then $G^*_{m,n}$ is the full clique on n nodes, minus the edge $(n-1, n)$.

An illustration of $\gamma(G^*_{m,n})$ as a function of the number of edges m is given in Fig. 5. One item to note is that there may be extra edges not used in our design of $G^*_{m,n}$. For instance, in Fig. 5, the efficiency is the same when $12 \leq m \leq 19$. This implies that $G^*_{12,10}$ and $G^*_{19,10}$ can be the same graph, and for any value of m in between. Hence, there are "dead zones" seen in the graph in Fig. 5.

C. Sibling Property

Here, we present a graph property, along with a corollary to Theorem 1. These results are key to the proof for Theorem 2.

Definition 1: Let $G \in \mathcal{G}$. Then, G has the *Sibling Property* if for some maximum independent set J, there exist $w \in V \setminus J$ and $i \in J$ such that $i \in \mathcal{N}_w$ (see Fig. 2).

Lemma 1: If a graph G lacks the Sibling Property, then
1) There is a unique maximum independent set J.
2) The set J must include nodes n and $n - 1$.
3) The induced subgraph G' created by removing the set J from G must be such that $\alpha(G) > \alpha(G')$.
4) Every node outside the set J must have outgoing edges to at least 2 nodes in J.[6]

Proof: We prove each Property separately.

Property 1: Suppose there are two maximum independent sets J and J'. Let $i \in J' \setminus J$ and $j \in J \setminus J'$. By definition, all nodes in either set cannot have any outgoing edges. This implies that $(i,j),(j,i) \notin E$: in other words, i and j are independent from each other, and neither J nor J' are maximum, a contradiction.

Property 2: First, suppose that n is not included in J. Then, there exists an edge (i,n) for some $i \in J$. By definition, this means G has the Sibling Property, a contradiction. Now suppose that $n - 1$ is not in J. Since G does not have the Sibling Property, then by definition, $(j, n-1) \notin E$ for all $j \in J \setminus n$. This means that another maximum independent set is $\{n-1\} \cup J \setminus n$, which is a contradiction to statement 1.

Property 3: If this were not true, then J would not be a unique maximum independent set.

Property 4: Let $i \notin J$. By definition, i cannot have any incoming edges from J and if there are no edges between i and J, then i must be part of J, a contradiction. Therefore, we consider the case where $(i,j) \in E$ for some $j \in J$, but no outgoing edges from i to $J \setminus j$ exist. This means that another maximum independent set is $i \cup J \setminus j$, and J is not unique. By Property 1, this is a contradiction. ∎

Corollary 1: For a graph $G \in \mathcal{G}$ with the Sibling Property

$$\gamma(G) \leq \frac{1}{1 + \alpha(G)} \tag{35}$$

with equality when $\alpha(G) = k(G)$.

Proof: We provide an example, which gives us the upper bound using a weighted set cover problem. Let J be a maximum independent set of G and let w be defined as in Definition 1. Then, $\mathcal{T} = \{t_1, \ldots, t_n\}$, where $v_i = 1$ if $i \in J$ or $i = w$, and $v_i = 0$ otherwise. The action sets are

$$X_i = \begin{cases} \{\{t_w\}, \{t_i\}\} & \text{if } i \in J \\ \{\{t_w\}\} & \text{if } i = w \\ \{\{t_i\}\} & \text{otherwise.} \end{cases} \tag{36}$$

Each agent in J is equally incentivized to choose either option, since none of them can have access to the choice of the others. Therefore, the worst case in the greedy algorithm is for every agent in J to choose t_w, implying $f(x^{\text{sol}}) = v_w = 1$. Each agent makes the other choice in the optimal, so $f(x^{\text{opt}}) = v_w + \sum_{i \in J} v_i = 1 + \alpha(G)$. Therefore, $\gamma(f, X, G) = 1/(1 + \alpha(G))$ is an upper bound on $\gamma(G)$.

In the case, where $\alpha(G) = k(G)$, (14) shows that $\alpha^*(G) = \alpha(G)$, which implies by Theorem 1 that $\gamma(G) \geq 1/(1 + \alpha(G))$. ∎

D. Proof for Theorem 2

In this section, we present the proof for Theorem 2, beginning with two lemmas. The first characterizes the number of edges in a complement Turán graph, and the second characterizes the fewest number of edges in a graph without the Sibling Property.

Lemma 2: Let $G = \overline{T(n,r)}$. Then, the number of edges in G is

$$M(n,r) := \frac{1}{2}(n \bmod r)\left\lceil \frac{n}{r} \right\rceil \left(\left\lceil \frac{n}{r} \right\rceil - 1\right)$$
$$+ \frac{1}{2}(r - n \bmod r)\left\lfloor \frac{n}{r} \right\rfloor \left(\left\lfloor \frac{n}{r} \right\rfloor - 1\right). \tag{37}$$

Proof: This can be shown by construction. Recall that $\overline{T(n,r)}$ is a set of disconnected cliques, of as close to equal size as possible. Making purely equal-sized cliques would mean that each clique is of size $\lfloor n/r \rfloor$, with $n \bmod r$ nodes left over. If each of these remaining nodes is added to a different clique, then G consists of $n \bmod r$ cliques of size $\lceil n/r \rceil$ and the rest of size $\lfloor n/r \rfloor$. Since a clique of size p contains $\frac{1}{2}p(p-1)$ edges, we see that the first line in (37) is the number of edges in all the larger cliques, and the second line is the number of edges in all the smaller cliques. ∎
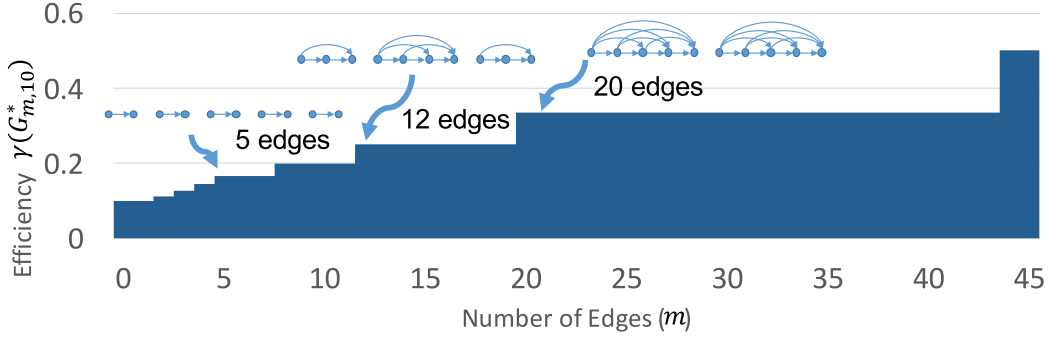
Fig. 5. Efficiency of $G^*_{m,10}$ for all values of m, with example graphs for a few values of m. Notice the "dead zones," where adding more edges does not lead to any higher efficiency guarantees.

Lemma 3: Let $G \in \mathcal{G}$ have n nodes, be without the Sibling Property, and such that $\alpha(G) = r$. Then, the number of edges m in G satisfies

$$m \geq M(n-r, r-1) + 2(n-r). \tag{38}$$

Furthermore, for any values of n and r, such a G can be constructed so that (38) is at equality.

Proof: In this proof, we construct a G such that (38) is at equality, then reason that no other graph with n nodes, independence number r, and without the Sibling Property can have fewer edges. The proof also leverages the properties for a graph without the Sibling Property, found in Lemma 1. Let J be the unique maximum independent set (Property 1) in G, and let G' be the induced subgraph of G created by removing the nodes in J. Then, we know that $\alpha(G') < \alpha(G)$ (Property 3). From (32) and Lemma 2, the minimum number of edges that such a G' can have is $M(n-r, r-1)$. Finally, every node in G' must have outgoing edges to at least two nodes in J, therefore, G must have an additional $2(n-r)$ edges. Thus, the minimum number of edges to construct G is given in (38). ∎

We now commence with the proof for Theorem 2. The case $m = 0$ trivially holds, so we assume that $m > 0$. Recall that the graph $\hat{T}(n,m)$ is a set of disconnected cliques, which implies that any maximum independent set has one node from each clique, and that no maximum independent set is unique. Therefore, by Lemma 1, Property 1, $\hat{T}(n,m)$ has the Sibling Property. In light of (33), it follows from Corollary 1 that $\gamma(\hat{T}(n,m)) = 1/(1 + \alpha(\hat{T}(n,m)))$. The statement in (32) also shows that no other graph with $\leq m$ edges can have a smaller independence number. Combining this with Corollary 1 implies that no other graph with the Sibling Property (and same number of nodes and edges) can have a higher efficiency.

It remains to confirm that any graph without the Sibling Property cannot have a higher efficiency than $\hat{T}(n,m)$, given n nodes and m edges—with the exception when $m = \frac{1}{2}n(n-1)$. Let G be a graph with n nodes, m edges, without the Sibling Property, and with independence number $r + 1$. We assume that G has the fewest number of edges (as dictated by Lemma 3), with the highest possible efficiency $\gamma(G) = 1/(r+1)$. By Corollary 1 and (33), this is the same efficiency as $\hat{T}(n,m)$, thus we seek to characterize when the number of edges in G is greater than or equal to that of $\hat{T}(n,m)$. In other words, $\mathcal{G}^*_{m,n} = \hat{T}(n,m)$

only if

$$M(n,r) \leq m = M(n-r-1, r) + 2(n-r-1). \tag{39}$$

In order to show when this condition holds, we divide the remainder of the proof into four cases, the union of which covers all possible values of n and r. In the first case, when $r = 1$, we prove (39) is false for all values of n (which corresponds to the case in the theorem statement when $m = \frac{1}{2}n(n-1) - 1$). In the other cases, we show that (39) is true, justifying that $G^*_{m,n} = \hat{T}(n,m)$.

Case 1. $r = 1$: Here, $\hat{T}(n,m)$ is a clique, and has $\frac{1}{2}n(n-1)$ edges. The graph G is such that

$$m = M(n-2, 1) + 2(n-2) = \frac{1}{2}n(n-1) - 1 \tag{40}$$

which is one less edge than $\hat{T}(n,m)$. Thus, for any value of n, there exists a G where (39) is false. Such a G is shown for $n = 4$ in Fig. 2(a), and a trivial extension to the proof in Appendix A shows that $\gamma(G) = \gamma(\hat{T}(n,m)) = 1/2$ for any value of n. Since G is created with the fewest number of edges, it follows that (39) is false *only* when $m = \frac{1}{2}n(n-1) - 1$. By the construction in the proof of Lemma 3, such a G is the full clique minus the edge $(n-1, n)$.

Case 2. $r = n - 1 \geq 2$: In this case, $\hat{T}(n,m)$ is the graph with no edges and efficiency $1/n$. Any graph with 1 or 0 edges must have this same efficiency, so (39) is true in this case.

In the remaining cases, we assume that $2 \leq r \leq n - 2$, which also implies that $n \geq 4$. In both cases, we show that (39) holds.

Case 3. $n \bmod r \geq 1$: This condition implies the following.
1) $(n-r-1) \bmod r = n \bmod r - 1$.
2) $\lceil n/r \rceil = \lfloor n/r \rfloor + 1$.
3) $\lfloor (n-r-1)/r \rfloor = \lfloor r/n \rfloor - 1$.
4) $\lceil (n-r-1)/r \rceil = \lfloor r/n \rfloor$.

Leveraging the above statements, $M(n,r)$ and $M(n-r-1, r)$ become

$$M(n,r) = \frac{1}{2} \left\lfloor \frac{n}{r} \right\rfloor \left(2(n \bmod r) + r \left\lfloor \frac{n}{r} \right\rfloor - r \right) \tag{41}$$

$$M(n-r-1, r) = (n \bmod r)\left(\left\lfloor \frac{n}{r} \right\rfloor - 1 \right) + \frac{r}{2}\left\lfloor \frac{n}{r} \right\rfloor^2 - 3\left\lfloor \frac{n}{r} \right\rfloor. \tag{42}$$

Using these expressions to evaluate (39) yields

$$\left\lfloor \frac{n}{r} \right\rfloor (r+1) + n \bmod r \le 2n - r - 1. \tag{43}$$

We can now use the identity $n \bmod r = n - r\lfloor n/r \rfloor$ to change the requirement in (43) to

$$\left\lfloor \frac{n}{r} \right\rfloor \le n - r - 1. \tag{44}$$

Since $\lfloor n/r \rfloor \le n/r$, a sufficient statement for (43) to hold can be found by replacing $\lfloor n/r \rfloor$ with n/r, which can be simplified to

$$\frac{r^2 + 1}{r - 1} \le n. \tag{45}$$

The expression on the left-hand side of the inequality is non-decreasing in r. Since $r \le n - 2$, if the inequality is true for $r = n - 2$, then it is true for all relevant values of n, r. If we let $r = n - 2$ in (45) and simplify, we conclude that (45) holds for all $n \ge 5$. By the premise that $n \ge 4$, the only values of n, r that could make this false are $n = 4, r = 2$; however, this would imply that $n \bmod r = 0$, not allowable by the condition for this case. Thus, (45) is true and by extension so is (39).

Case 4. $n \bmod r = 0$: The condition implies the following.
1) $(n - r - 1) \bmod r = r - 1$.
2) $\lfloor n/r \rfloor = \lceil n/r \rceil = n/r$.
3) $\lfloor (n - r - 1)/r \rfloor = \lfloor r/n \rfloor - 2$.
4) $\lceil (n - r - 1)/r \rceil = \lfloor r/n \rfloor - 1$.

Leveraging the above statements, $M(n, r)$ and $M(n - r - 1, r)$ become

$$M(n, r) = \frac{n}{2}\left(\frac{n}{r} - 1\right) \tag{46}$$

$$M(n - r - 1, r) = \frac{n^2}{2r} - \frac{3}{2}n + r - \frac{n}{r} + 2. \tag{47}$$

Using these expressions to evaluate (39) yields

$$\frac{r^2}{r - 1} \le n. \tag{48}$$

It is straightforward to see that if (45) holds, then so does (48). Case 3 shows that (45) is true unless $r = 2, n = 4$. However, (48) is true for these values, which implies (48) holds for all relevant values of n, r. Thus, in this case, (39) also holds. $\square$

VI. CONCLUSION

We have derived bounds on the worst-case efficiency of the distributed greedy algorithm for submodular maximization. These bounds can be used to design information sharing constraints that maximize the worst-case efficiency.

Future research can follow in several directions. For example, while the bounds presented in Theorem 1 are applicable to any graph structure $G \in \mathcal{G}$, it is not immediately clear how to characterize graphs where $\gamma(G) = 1/\alpha^*(G)$ versus those where $\gamma(G) = 1/(\alpha^*(G) + 1)$, and if there exist graphs where $\gamma(G)$ is in between. Precisely defining $\gamma(G)$ in terms of its structural properties is another possibility.

Another idea for future research is to consider a game-theoretic approach instead of a greedy approach, similar to work

done in [32]. In this case, the order of the agents would not matter, and the goal would be to characterize the Nash equilibrium for certain graph structures. Finally, future work could include exploring the use of a different utility function rather than marginal contribution in order to make greedy decisions.

APPENDIX

A. Proof for Lower Bound in Example 4

We begin with the following inequality:

$$f(x^{\mathrm{opt}}) \le f(x^{\mathrm{opt}}, x_{1:2}) \tag{49}$$

$$= f(x_{1:2}) + \Delta(x_1^{\mathrm{opt}}|x_{1:2}) + \Delta(x_2^{\mathrm{opt}}|x_1^{\mathrm{opt}}, x_{1:2})$$
$$+ \Delta(x_3^{\mathrm{opt}}|x_{1:2}^{\mathrm{opt}}, x_{1:2}) + \Delta(x_4^{\mathrm{opt}}|x_{1:3}^{\mathrm{opt}}, x_{1:2}) \tag{50}$$

$$\le f(x_{1:2}) + f(x_1^{\mathrm{opt}}) + \Delta(x_2^{\mathrm{opt}}|x_1) + \Delta(x_3^{\mathrm{opt}}|x_{1:2})$$
$$+ \Delta(x_4^{\mathrm{opt}}|x_{1:2}) \tag{51}$$

$$\le f(x_{1:2}) + f(x_1) + \Delta(x_2|x_1) + \Delta(x_3|x_{1:2})$$
$$+ \Delta(x_4|x_{1:2}) \tag{52}$$

$$= f(x_{1:3}) + f(x_{1:2}, x_4) \le 2f(x_{1:4}) \tag{53}$$

where (49) is true by submodularity, (50) is true by definition of Δ, (51) is true by submodularity, (52) is true since agents choose according to (7), and (53) is true by definition of Δ and submodularity. Therefore, we see $\gamma(G) = 1/2$. $\square$

REFERENCES

[1] D. Grimsman, M. S. Ali, J. P. Hespanha, and J. R. Marden, "Impact of information in greedy submodular maximization," in *Proc. IEEE 56th Annu. Conf. Decision Control*, 2017, pp. 2900–2905.

[2] A. Krause and C. Guestrin, "Near-optimal observation selection using submodular functions," in *Proc. Assoc. Adv. Artif. Intell.*, 2007, vol. 7, pp. 1650–1654.

[3] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proc. SIGKDD Conf. Knowl. Discovery Data Mining*, 2003, pp. 137–146.

[4] P. Kohli, M. P. Kumar, and P. H. Torr, "P^3 & beyond: Move making algorithms for solving higher order functions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 9, pp. 1645–1656, Sep. 2009.

[5] O. Barinova, V. Lempitsky, and P. Kholi, "On detection of multiple object instances using Hough transforms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1773–1784, Sep. 2012.

[6] H. Lin and J. Bilmes, "A class of submodular functions for document summarization," in *Proc. 49th Annu. Meet. Assoc. Comput. Linguistics: Human Lang. Technol.*, 2011, pp. 510–520.

[7] A. Singh, A. Krause, C. Guestrin, W. J. Kaiser, and M. A. Batalin, "Efficient planning of informative paths for multiple robots," in *Proc. Int. Joint Conf. Artif. Intell.*, 2007, vol. 7, pp. 2204–2211.

[8] A. Krause, R. Rajagopal, A. Gupta, and C. Guestrin, "Simultaneous placement and scheduling of sensors," in *Proc. Int. Conf. Inf. Process. Sensor Netw.*, 2009, pp. 181–192.

[9] J. R. Marden, "The role of information in distributed resource allocation," *IEEE Trans. Control Netw. Syst.*, vol. 4, no. 3, pp. 654–664, Sep. 2017.

[10] M. Grötschel, L. Lovász, and A. Schrijver, "The ellipsoid method and its consequences in combinatorial optimization," *Combinatorica*, vol. 1, no. 2, pp. 169–197, 1981.

[11] S. Iwata, L. Fleischer, and S. Fujishige, "A combinatorial strongly polynomial algorithm for minimizing submodular functions," *J. ACM*, vol. 48, no. 4, pp. 761–777, 2001.

[12] A. Schrijver, "A combinatorial algorithm minimizing submodular functions in strongly polynomial time," *J. Combinatorial Theory, Series B*, vol. 80, no. 2, pp. 346–355, 2000.

[13] L. Lovász, "Submodular functions and convexity," in *Mathematical Programming: The State of the Art*. New York, NY, USA: Springer, 1983, pp. 235–257.

[14] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions I," *Math. Program.*, 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 *Polyhedral Combinatorics*. New York, NY, USA: Springer, 1978, pp. 73–87.

[16] M. Minoux, "Accelerated greedy algorithms for maximizing submodular set functions," in *Optimization Techniques*. New York, NY, USA: Springer, 1978, pp. 234–243.

[17] N. Buchbinder, M. Feldman, J. Seffi, and R. Schwartz, "A tight linear time (1/2)-approximation for unconstrained submodular maximization," *SIAM J. Comput.*, vol. 44, no. 5, pp. 1384–1402, 2015.

[18] J. Vondrák, "Optimal approximation for the submodular welfare problem in the value oracle model," in *Proc. ACM Symp. Theory Comput.*, 2008, pp. 67–74.

[19] M. Sviridenko, "A note on maximizing a submodular set function subject to a knapsack constraint," *Operations Res. Lett.*, vol. 32, no. 1, pp. 41–43, 2004.

[20] G. Qu, D. Brown, and N. Li, "Distributed greedy algorithm for multi-agent task assignment problem with submodular utility functions," Jun. 2018. [Online]. Available: https://nali.seas.harvard.edu/files/nali/files/submodular_automatica2018.pdf

[21] G. Calinescu, C. Chekuri, M. Pál, and J. Vondrák, "Maximizing a submodular set function subject to a matroid constraint," in *Proc. Int. Conf. Integer Program. Combinatorial Optim.*, 2007, pp. 182–196.

[22] Y. Filmus and J. Ward, "The power of local search: Maximum coverage over a matroid," in *Proc. Symp. Theor. Aspects Comput. Sci.*, 2012, vol. 14, pp. 601–612.

[23] U. Feige, "A threshold of ln n for approximating set cover," *J. ACM*, vol. 45, no. 4, pp. 634–652, 1998.

[24] B. Mirzasoleiman, A. Karbasi, R. Sarkar, and A. Krause, "Distributed submodular maximization: Identifying representative elements in massive data," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 2049–2057.

[25] B. Gharesifard and S. L. Smith, "Distributed submodular maximization with limited information," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 4, pp. 1635–1645, Dec. 2018.

[26] G. Arslan, J. R. Marden, and J. S. Shamma, "Autonomous vehicle-target assignment: A game-theoretical formulation," *J. Dyn. Syst. Meas. Control*, vol. 129, no. 5, pp. 584–596, 2007.

[27] M. Gairing, "Covering games: Approximation through non-cooperation," in *Proc. Int. Workshop Internet Netw. Econ.*, 2009, pp. 184–195.

[28] C. Godsil and G. F. Royle, *Algebraic Graph Theory*, vol. 207. New York, NY, USA: Springer, 2013.

[29] S. Arumugam and K. R. Kumar, "Fractional independence and fractional domination chain in graphs," *AKCE Int. J. Graphs Combinatorics*, vol. 4, no. 2, pp. 161–169, 2007.

[30] P. Turán, "On an extremal problem in graph theory," *Mat. Fiz. Lapok*, vol. 48, pp. 436–452, 1941.

[31] L. Volkmann, "On perfect and unique maximum independent sets in graphs," *Mathematica Bohemica*, vol. 129, no. 3, pp. 273–282, 2004.

[32] A. Vetta, "Nash equilibria in competitive societies, with applications to facility location, traffic routing and auctions," in *Proc. Symp. Found. Comput. Sci.*, 2002, pp. 416–425.

Mohd. Shabbir Ali received the B.Eng. degree in electrical and electronics engineering from the MuffakhamJah College of Engineering and Technology, Hyderabad, India, in 2010. He received the Ph.D. degree in computer science and networks from Telecom ParisTech, Paris, France, in 2017. He received the M.Sc. degree in electrical communication engineering from Indian Institute of Science, Bangalore, India, in 2014.

He is a Postdoctoral Researcher with Orange Labs, Paris, France. His research interests include game theory, machine learning, distributed optimization, heterogeneous cellular networks, cognitive radio networks, resource allocation, load balancing, and interference modeling and management.

João P. Hespanha was born in Coimbra, Portugal, in 1968. He received the Licenciatura degree in electrical and computer engineering from the Instituto Superior Técnico, Lisbon, Portugal, in 1991 and the Ph.D. degree in electrical engineering and applied science from Yale University, New Haven, CT, USA, in 1998.

From 1999 to 2001, he was an Assistant Professor with the University of Southern California, Los Angeles, CA, USA. He moved to the University of California, Santa Barbara, Santa Barbara, CA, USA in 2002, where he currently holds a Professor position with the Department of Electrical and Computer Engineering. His current research interests include hybrid and switched systems, multiagent control systems, game theory, optimization, distributed control over communication networks (also known as networked control systems), the use of vision in feedback control, stochastic modeling in biology, and network security.

Dr. Hespanha is the recipient of Yale University's Henry Prentiss Becton Graduate Prize for exceptional achievement in research in Engineering and Applied Science, a National Science Foundation CAREER Award, the 2005 best paper award at the second International Conference on Intelligent Sensing and Information Processing, the 2005 Automatica Theory/Methodology best paper prize, the 2006 George S. Axelby Outstanding Paper Award, and the 2009 Ruberti Young Researcher Prize. He is a Fellow of the International Federation of Automatic Control. He was an IEEE Distinguished Lecturer from 2007 to 2013.

David Grimsman received the B.S. degree in electrical and computer engineering from Brigham Young University, Provo, UT, USA, in 2006 as a Heritage Scholar, with a focus on signals and systems. He received the M.S. degree in computer science from Brigham Young University in 2016. He is currently working toward the Ph.D. degree with the Electrical and Computer Engineering Department, University of California, Santa Barbara, Santa Barbara, CA, USA.

He worked with BrainStorm, Inc. for several years as a Trainer and IT Manager. His research interests include multiagent systems, game theory, distributed optimization, network science, linear systems theory, and security of cyberphysical systems.

Jason R. Marden received the B.S. and Ph.D. degrees in mechanical engineering (under the supervision of Jeff S. Shamma), from the University of California, Los Angeles, CA, USA, in 2001 and 2007, respectively.

He is an Assistant Professor with the Department of Electrical and Computer Engineering, University of California, Santa Barbara, Santa Barbara, CA, USA. After graduating, he was a Junior Fellow with the Social and Information Sciences Laboratory, California Institute of Technology, Pasadena, CA, USA, until 2010 and then an Assistant Professor with the University of Colorado, Boulder, CO, until 2015. His research interests focus on game-theoretic methods for the control of distributed multiagent systems.

Dr. Marden is a recipient of an ONR Young Investigator Award (2015), an NSF Career Award (2014), the AFOSR Young Investigator Award (2012), the SIAM CST Best Sicon Paper Award (2015), and the American Automatic Control Council Donald P. Eckman Award (2012). He received the Outstanding Graduating Ph.D. Student award in mechanical engineering from the University of California, Los Angeles.