

Impact of Information in Greedy Submodular Maximization

David Grimsman*, Mohd. Shabbir Ali*, João P. Hespanha and Jason R. Marden

Abstract—The maximization of submodular functions is a well-studied topic due to its application in many common engineering problems. Because this problem has been shown to be NP-Hard for certain subclasses of functions, much work has been done to develop efficient algorithms to approximate an optimal solution. Among these is a simple greedy algorithm, which has been shown to guarantee a solution within $\frac{1}{2}$ the optimal. However, when this algorithm is implemented in a distributed way, it requires all agents to share information with one another - a costly constraint for some applications. This work explores how the degradation of information sharing among the agents affects the performance of the distributed greedy algorithm. For any underlying communication graph structure, we show results for how well the distributed greedy algorithm can perform. In addition, for applications where the number of agents and number of communication links is fixed, we identify near-optimal graph structures with the highest performance guarantees. This result can inform system designers as to the most impactful places to insert communication links.

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 [9], maximizing influence (viral marketing) in social networks [7], image segmentation in image processing [8], multiple object detection in computer vision [1], document summarization [11], path planning of multiple robots [19], sensor placement [10], [14], and resource allocation in multi-agent systems [14]. For each case, the problem can be formulated as the optimization of a submodular function.

While polynomial algorithms exist to solve submodular minimization, [5], [6], [18], maximization has been shown to be NP-Hard for certain subclasses of submodular functions [12]. Thus a tremendous effort has been placed on developing fast algorithms that approximate the solution to the submodular maximization problem [16], [3], [15], [2], [23], [20], [17]. The resounding message from this extensive research is that very simple algorithms can provide strong guarantees on the quality of the approximation. Specifically, this work focuses on submodular maximization with a matroid constraint. Using a greedy randomized approach, [23]

shows that the optimal solution can be approximated within a factor of $1 - 1/e \approx 0.63$.

Another proposed technique is to use a simple greedy algorithm, which has been shown to guarantee an approximate solution within $1/2$ of the optimal [16]. One reason this algorithm remains relevant is that it lends itself to a distributed approach to maximization. In this approach, the distributed greedy algorithm, a set of agents sequentially maximizes a function without the need of centralized information. The result is simply the compilation of all the agents' decisions. As each agent maximizes its function, however, it must be able to evaluate all its potential choices and have access to the decisions of all previous agents. In many multi-agent systems, this amount of information and sharing is costly.

Research has therefore begun to explore how limited information and sharing can impact the performance of the distributed greedy algorithm. For example, [13] explores the performance when an agent can only evaluate a local subset of its choices. The work in [4] studies the performance when and agent can only observe a local subset of its predecessors. In each case, results show that localizing information can substantially degrade performance.

This paper more closely relates to the work done in [4] in evaluating information sharing constraints. Their work models communication structures as graphs and demonstrates how performance can degrade for certain graphs. For instance, if agents are partitioned into groups that only transmit among themselves, the lower bound on performance degrades by a factor proportional to the number of groups. However, two questions remain open:

- 1) For any communication structure, what are the performance guarantees?
- 2) What communication structures make the highest performance guarantees?

The contribution of this paper is to provide insights into answering these two questions. More specifically

- 1) Theorem 1 gives lower and upper bounds on worst-case performance for a given communication structure. The bounds give intuition on what properties of the underlying graph need to be improved in order for performance to improve.
- 2) Theorem 2 shows the best performance that a system designer can achieve with a fixed number of agents and communication links. The results show that when information is costly, the best communication structures spread out the communication links among the agents, rather than clustering them among a small group.

The remainder of this paper is dedicated to proving these

*These authors contributed equally to this work.

D. Grimsman (davidgrimsman@umail.ucsb.edu), J. R. Marden (jmarden@ece.ucsb.edu), and J. P. Hespanha (hespanha@ece.ucsb.edu) are with the Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA . Corresponding author.

M. S. Ali is with LTCI, Telecom ParisTech, Université Paris-Saclay. He was sponsored by NetLearn ANR project (ANR-13-INFR-004). mdshabbirali88@gmail.com.

This work is supported by ONR Grant #N00014-17-1-2060 and NSF Grant #ECCS-1638214.

two theorems.

II. MODEL

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

- *Submodular*: For $A \subseteq B \subset S$ and $x \in S \setminus B$, the following holds:

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

- *Monotonic*: For $A \subseteq B$, $f(A) \leq f(B)$.
- *Normalized*: $f(\emptyset) = 0$.

We say f is *modular* if (1) is an equality.

In many applications maximizing such f can be formulated as the compilation of decisions made by agents, each maximizing a local utility. Suppose we have agents labeled $1, \dots, n$ and a corresponding family of sets $X_1, \dots, X_n$, where $X_i \subseteq 2^S$. Here we define $X = X_1 \times \dots \times X_n$. For notational purposes, if A is a set of agents, then let $x_A = \bigcup_A x_i$, where $x_i \in X_i$ and $i \in A$. We also use the notation that $x_{a:b} = \bigcup_{i=a}^b x_i$ for $b \geq a$. Finally, we define $x_1^*, \dots, x_n^*$ as the optimal set of decisions for each agent. In this context, the submodular maximization problem can be formulated as

$$f(x_{1:n}^*) = \max_{x_1 \in X_1, \dots, x_n \in X_n} f(x_{1:n}). \quad (2)$$

This type of constraint where we choose from a family of subsets X_i is equivalent to a partition matroid constraint [4]. Each X_i can be considered the action space of agent i .

As mentioned in the previous section, the maximization problem in (2) can be approximated using a distributed greedy algorithm. Since the algorithm requires agents to make decisions sequentially, without loss of generality we impose an ordering on the agents according to their labels, i.e., agent 1 chooses first, agent 2 chooses second, etc. Agent i makes its choice $x_i \in X_i$ based on the following rule:

$$x_i \in \arg \max_{\tilde{x}_i \in X_i} f(\tilde{x}_i \cup x_{1:i-1}), \quad (3)$$

where $x_{1:0}$ is defined to be $\emptyset$. Note that there could be several x_i to choose from in the $\arg \max$, so define $\mathcal{X} \subseteq X$ to be the set of all possible choices if agents choose according to (3). It is well-known in the literature that for any submodular f , any set of action spaces X , and any order of agents, the quality of the resulting solution $x = x_{1:n}$ derived from the distributed greedy algorithm compared to the optimal solution $x^* = x_{1:n}^*$ satisfies

$$\gamma(f, X) := \frac{\inf_{x \in \mathcal{X}} f(x)}{f(x^*)} \geq \frac{1}{2}, \quad (4)$$

which means the solution is within 50% of the optimal [16]. Here we say that $\gamma(f, X)$ is the efficiency of solution x . For special classes of submodular functions, and additional constraints placed on $X_1, \dots, X_n$, [16] also shows that $\gamma(f, X) \geq 1 - 1/e$.

Notice that in (3), agent i must have access to the decisions of all previous agents. However, this may not hold in

some real-world applications. Therefore a more generalized version of the distributed greedy algorithm is where agent i makes its choice using the following rule:

$$x_i \in \arg \max_{\tilde{x}_i \in X_i} f(\tilde{x}_i \cup x_{\mathcal{N}_i}), \quad (5)$$

where $\mathcal{N}_i \subseteq \{1, \dots, i-1\}$. The sets $\mathcal{N}_i$, $i = 1, \dots, n$ are the sets that represent the communication structure - in other words, $\mathcal{N}_i$ is the set of agents whose choices are observed by agent i when making its decision. It is helpful to visualize this communication structure as a graph $G = (V, E)$, where V is a set of vertices and $E \subseteq V \times V$ is a set of edges between vertices. In this scenario each vertex is an agent and each edge (j, i) implies $j \in \mathcal{N}_i$, 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

$$\mathcal{G} = \{G = (V, E) : (i, j) \in E \implies i < j\} \quad (6)$$

is the set of admissible graphs that correspond to a communication structure. In this context, we define $\mathcal{X} \subseteq X$ to be the set of all possible choices if agents choose according to (5), and again measure the efficiency of solution x according to the decision rule (5) with

$$\gamma(f, X, G) := \frac{\inf_{x \in \mathcal{X}} f(x)}{f(x^*)}, \quad (7)$$

where $G \in \mathcal{G}$ is defined by the sets $\mathcal{N}_i$.

The goal of this paper will be to identify the efficiency guarantees associated with this more generalized version of the distributed greedy algorithm for any monotone normalized submodular function. To that end, let $\mathcal{F}$ be the set of such functions, where each $f \in \mathcal{F}$ is defined on some element set S_f . We now define

$$\gamma(G) = \inf_{f \in \mathcal{F}} \inf_{X: X_i \subseteq 2^{S_f}} \gamma(f, X, G) \quad (8)$$

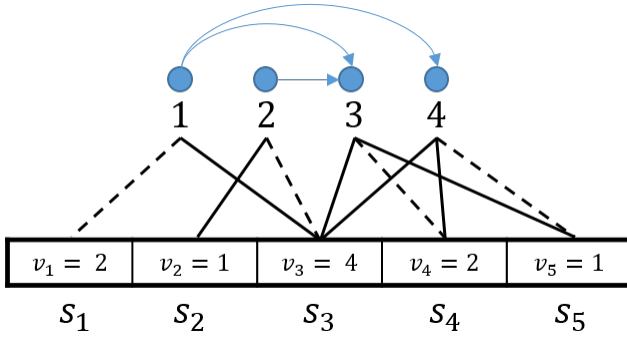
In words, $\gamma(G)$ is the worst-case efficiency for any f and family of sets $X_1, \dots, X_n$ as defined above, given that the communication structure among the agents is represented by G .

A. An Illustrative Example: Weighted Set Cover

We now describe the weighted set cover problem using our model, in order to illustrate how the generalized distributed greedy algorithm works using a communication structure represented by a graph G . The following sections leverage instances of this problem to show worst-case scenarios and to prove tight lower bounds, so it serves as both a simple example and part of our proof technique. For a base set of elements S :

- let $s_1, \dots, s_T$ be a partition on S , each with an associated value $v_1, \dots, v_T$,
- let $X_i \subseteq \{s_1, \dots, s_T\}$, and
- let $f(x_{1:n}) = \sum_{j \in J} v_j$, where $j \in J$ if s_j was chosen by some agent.

In this case, f is submodular (more specifically, it is modular), monotone, and normalized, thus it meets the requirements for the model. Essentially, the agents are collectively



(a) The setup of a weighted set cover problem. The set S is partitioned into subsets $s_1, \dots, s_5$, each with a corresponding value $v_1, \dots, v_5$. The available choices to each agent are represented by the black lines, where the dotted lines are an optimal set of choices. The goal for the agents is to maximize the value of the union their choices.

Algorithm	x_1	x_2	x_3	x_4	$f(x_{1:4})$
Optimal	s_2	s_3	s_4	s_5	9
Distributed Greedy	s_3	s_1	s_4	s_5	8
Generalized Distributed Greedy	s_3	s_3	s_4	s_4	6

(b) This table shows the agents' decisions in an optimal case, the case where the distributed greedy algorithm is used (agents choose according to (3)) and the case where the generalized distributed algorithm is used (agents choose according to (5), constrained to the graph shown).

Fig. 1: An instance of the weighted set cover problem

trying to “cover” as much of S as they can, given each agent i 's restriction to choose from X_i . An instance of the weighted cover set problem is shown in Figure 1.

III. WORST-CASE EFFICIENCY BOUNDS

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

A. Preliminaries

In order to show bounds on $\gamma(G)$, we leverage terminology and work done in graph theory. For all definitions in this section, we assume that $G = (V, E)$ is any general 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 size of the largest clique in G .
- A *clique cover* is a partition on V such that the nodes in each partition form a clique.
- The *clique cover number* $k(G)$ is the minimum number of sets within any clique cover of G .

As an example, consider the graph in Figure 2. Here there are 4 cliques of size 1 (one for each node), 5 cliques of size 2 (one representing each edge), and 2 cliques of size 3 (the sets $\{1, 2, 3\}$ and $\{1, 2, 4\}$). Thus the clique number is 3. A

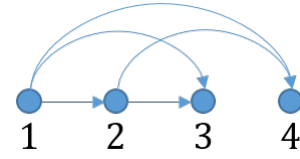


Fig. 2: An example graph used to illustrate some basic graph theory concepts. In this case $\omega(G) = 3$, $k(G) = 2$, and $\alpha(G) = 2$.

minimum clique cover is $\{1, 3\}, \{2, 4\}$, so the clique cover number is 2.

Another important notion in graph theory is that of independence:

- An *independent set* $I \subseteq V$ is a set of vertices such that $v_1, v_2 \in I$ implies $(v_1, v_2), (v_2, v_1) \notin E$.
- A *maximum independent set* $I_{\max}$ is an independent set of G such that no other independent set has more vertices.
- The *independence number* of G is $\alpha(G) := |I_{\max}|$.

As an example, consider again the graph in Figure 2. The maximum independent set is $\{3, 4\}$, thus the independence number is 2.

B. Related Work

As mentioned in the previous section, this work relates to that done in [4]. We give a brief description of their results, as it pertains to this section. The first is that for a graph $G \in \mathcal{G}$:

$$\gamma(G) \geq \frac{1}{n - \omega(G) + 2}, \quad (9)$$

and a second is that if G is a family of disconnected cliques, then

$$\gamma(G) \geq \frac{1}{2k(G)}. \quad (10)$$

The lower bound on $\gamma(G)$ that we show in the next section is greater than or equal to these bounds in all cases.

C. Lower and Upper Bounds on $\gamma(G)$

Now we present lower and upper bounds on $\gamma(G)$ in the terms of the clique cover number $k(G)$ and independence number $\alpha(G)$, respectively. In order to assist, let

$$\Delta(x_i | x_P) = f(x_i, x_P) - f(x_P) \quad (11)$$

be the marginal contribution of agent i given the choices of the agents in set P . We will also make use of the following Lemma:

Lemma 1: Let $A \subseteq S$ and $B \subseteq V$. Then

$$f(A, x_B) = f(A) + \sum_{i \in B} \Delta(x_i | A, x_{j \in B: j < i}). \quad (12)$$

Proof: Let j be the smallest element of B . From (11), it follows that $f(A, x_B) = f(A, x_{B \setminus \{j\}}, x_j) = f(A, x_{B \setminus \{j\}}) + \Delta(x_j | A, x_{B \setminus \{j\}})$. Let l be the next smallest element of B , then it follows that

$$f(A, x_B) = f(A, x_{B \setminus \{j, l\}}) + \Delta(x_j | A, x_{B \setminus \{j\}}) + \Delta(x_l | A, x_{B \setminus \{j, l\}}). \quad (13)$$

Continuing this line of reasoning sequentially for all $i \in B$, we arrive at (12) ■

We now proceed with the main theorem and its proof.

Theorem 1: Let $G \in \mathcal{G}$. Then

$$\frac{1}{\alpha(G)} \geq \gamma(G) \geq \frac{1}{k(G) + 1}. \quad (14)$$

Proof: We first begin to show the lower bound begin with the following inequality:

$$f(x_{1:n}^*) \leq f(x_{1:n}^*, x_{1:n}), \quad (15)$$

$$= f(x_{1:n}) + \sum_{i=1}^n \Delta(x_i^* | x_{1:n}, x_{1:i-1}^*), \quad (16)$$

$$\leq f(x_{1:n}) + \sum_{i=1}^n \Delta(x_i^* | \mathcal{N}_i), \quad (17)$$

$$\leq f(x_{1:n}) + \sum_{i=1}^n \Delta(x_i | \mathcal{N}_i), \quad (18)$$

where (15) holds by submodularity, (16) holds by Lemma 1, (17) holds by submodularity since $\mathcal{N}_i \subseteq \{1, \dots, n\}$, and (18) holds because agents make decisions according to (5).

Now let $C_1, \dots, C_k$ be a minimal clique cover of G . In other words, $C_1, \dots, C_k$ is a partition on V where for every $j \in \{1, \dots, k\}$ every vertex in C_j is connected to every other vertex in C_j . Let P be the function that maps vertex i to its assigned partition, i.e., if $i \in C_j$, then $P(i) = j$. We also define $Q_i = \{m \in C_{P(i)} : m < i\}$. Now we see that

$$\sum_{i=1}^n \Delta(x_i | \mathcal{N}_i) \leq \sum_{i=1}^n \Delta(x_i | x_{Q_i}), \quad (19)$$

$$= \sum_{j=1}^{k(G)} \sum_{i \in C_j} \Delta(x_i | x_{Q_i}), \quad (20)$$

$$= \sum_{j=1}^{k(G)} f(x_{C_j}), \quad (21)$$

$$\leq \sum_{j=1}^{k(G)} f(x_{1:n}), \quad (22)$$

$$= k(G) f(x_{1:n}), \quad (23)$$

where (19) is true by submodularity since $Q_i \subseteq \mathcal{N}_i$, (20) is true since it merely imposes a different order on the sum, (21) is true by Lemma 1, (22) is true by submodularity since $C_j \subseteq \{1, \dots, n\}$, and (23) is the result of the sum. Substituting the above value into (18), we see that

$$f(x_{1:n}^*) \leq (k(G) + 1) f(x_{1:n}), \quad (24)$$

which holds for any f , S , and $X_1, \dots, X_n$. Therefore $\gamma(G) \geq \frac{1}{k(G)+1}$.

Next we prove the upper bound. It is sufficient to show that for any G we can choose S , f and $X_1, \dots, X_n$ such that $\gamma(f, X, G) = \frac{1}{\alpha(G)}$. Then by definition $\gamma(G)$ cannot be greater than $\gamma(f, X, G)$. Consider a weighted set cover problem where S is partitioned by $s_0, s_1, \dots, s_{\alpha(G)+1}$, where $v_0 = 0$, and where $v_1 = \dots = v_{\alpha(G)+1} = 1$. Suppose that

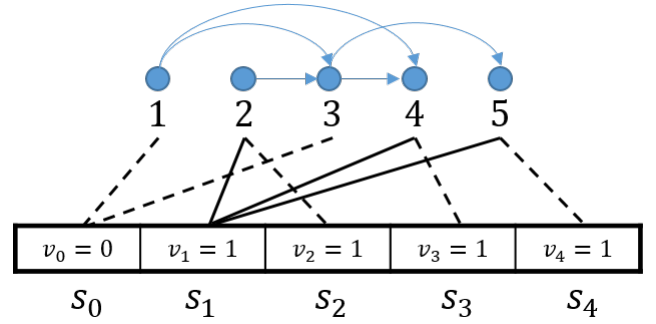


Fig. 3: An example weighted set cover problem from the proof of Theorem 1. Here we use $I_{\max} = \{2, 4, 5\}$. The optimal choices are shown by dotted lines, thus $f(x_{1:5}^*) = 3$. The worst case using the generalized distributed greedy algorithm, is where 2, 4, and 5 all choose s_1 , where $f(x_{1:5}) = 1$. Therefore, in this case $\gamma(f, X, G) = \frac{1}{\alpha}$.

$I_{\max}$ is a maximum independent set in G , and that the action sets are assigned as follows

$$X_i = \begin{cases} \{s_1, P(i)\} & \text{if } i \in I_{\max} \\ \{s_0\} & \text{otherwise,} \end{cases}$$

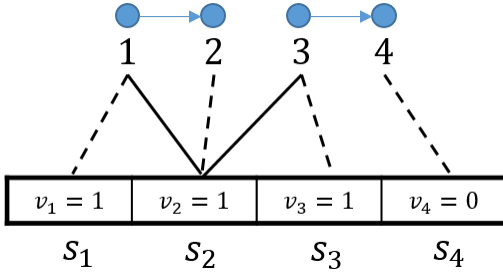
where $P : I_{\max} \rightarrow \{s_2, \dots, s_{\alpha(G)+1}\}$ is injective. An instance of this scenario is shown in Figure 3. Essentially, this choice of X_i has “zeroed out” any vertex that is not in $I_{\max}$, meaning that only vertices in $I_{\max}$ are adding value. However, note that by definition no vertex in $I_{\max}$ is connected to any other, so the corresponding agents make choices independently. The optimal set of choices are for each agent $i \in I_{\max}$ to choose $P(i)$, and thus $f(x_{1:n}^*) = \alpha(G)$. The worst set of choices would be for each agent $i \in I_{\max}$ to choose s_1 , which they have equal incentive to do. In this case, $f(x_{1:n}) = 1$, thus $\gamma(f, X, G) = \frac{1}{\alpha(G)}$. ■

As stated, the lower bound presented in Theorem 1 is greater than or equal to those lower bounds shown in [4]. This is trivially true for (10), and it’s true for (9) if $n - \omega(G) + 1 \geq k(G)$ for all $G \in \mathcal{G}$. We can see that this statement holds since $n - \omega(G)$ is all vertices outside a largest clique and $k(G)$ would not include any two from the same clique.

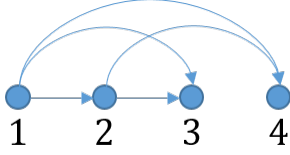
D. Examples

Theorem 1 shows lower and upper bounds on $\gamma(G)$, but we have not shown whether either of these bounds is tight. In fact, there exist some choices of f , S , and $X_1, \dots, X_n$ where the lower bound is tight and other choices where the upper bound is tight. In this section, we provide an example of each.

Example 1: Consider the weighted set coverage problem presented in Figure 4a. For this graph G , $\alpha(G) = k(G) = 2$. As shown, $\gamma(f, X, G) = \frac{1}{k(G)+1}$. Since $\gamma(G) \leq \gamma(f, X, G)$ by definition, it follows that $\gamma(G) = \frac{1}{k(G)+1}$. Also, since $\alpha(G) = 2$, the upper bound in Theorem 1 is $\frac{1}{2}$, and is not tight in this case. As a note, the lower bounds on $\gamma(G)$ in (9) and (10) are both $\frac{1}{4}$.



(a) An example of a graph G where $\gamma(G) = \frac{1}{k(G)+1}$. Here $k(G) = 2$, and we can see that $f(x_{1:4}^*) = 3$. The worst-case results from the generalized distributed greedy algorithm occur when $x_1 = x_2 = x_3 = s_2$, and therefore $f(x_{1:4}) = 1$. Therefore $\gamma(f, X, G) = \frac{1}{k(G)+1} = \frac{1}{3}$, so the lower bound in Theorem 1 is tight for this graph.



(b) An example of a graph where $\gamma_{min} = \frac{1}{\alpha}$

Fig. 4: Example graphs that meet the (a) lower bound and (b) upper bound given in Theorem 1.

Example 2: Consider the graph G in Figure 4b. Again $k(G) = \alpha(G) = 2$, and we claim the upper bound given in Theorem 1 is tight. This can be shown beginning with the following inequality:

$$f(x_{1:4}^*) \leq f(x_{1:4}^*, x_{1:2}) \quad (25)$$

$$= f(x_{1:2}) + \Delta(x_1^* | x_{1:2}) + \Delta(x_2^* | x_1^*, x_{1:2}) + \Delta(x_3^* | x_{1:2}, x_{1:2}) + \Delta(x_4^* | x_{1:3}^*, x_{1:2}) \quad (26)$$

$$\leq f(x_{1:2}) + f(x_1^*) + \Delta(x_2^* | x_1) + \Delta(x_3^* | x_{1:2}) + \Delta(x_4^* | x_{1:2}) \quad (27)$$

$$\leq f(x_{1:2}) + f(x_1) + \Delta(x_2 | x_1) + \Delta(x_3 | x_{1:2}) + \Delta(x_4 | x_{1:2}) \quad (28)$$

$$= f(x_{1:3}) + f(x_{1:2}, x_4) \quad (29)$$

$$\leq 2f(x_{1:4}), \quad (30)$$

where (25) is true by submodularity, (26) is true by Lemma 1, (27) is true by submodularity, (28) is true since agents choose according to (5), (29) is true by Lemma 1, and (30) is true by submodularity. Therefore we conclude $\gamma(G) = \frac{1}{2}$, which is the upper bound given in Theorem 1. We also see that since $k(G) = 2$, the lower bound from Theorem 1 is $\frac{1}{3}$ and is therefore not tight.

IV. OPTIMAL GRAPH STRUCTURES

The bounds on $\gamma(G)$ given above dictate a near-optimal approach to communication structure design. Using these results, we describe how to build graph structures that approximate the highest $\gamma(G)$. In other words, given n vertices and m edges, we show where to put edges in order

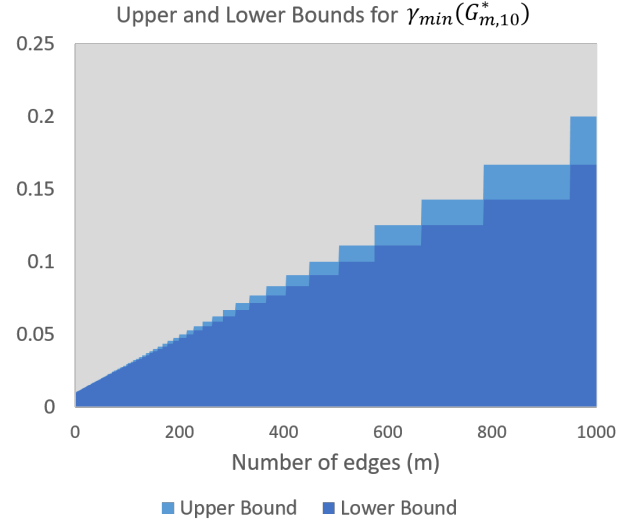


Fig. 5: The lower and upper bounds shown in Theorem 2 for graphs with 10 vertices, ranging from 0 to 1000 edges.

to maximize $\gamma(G)$. We begin by defining the following notation:

- Let $\mathcal{G}_{m,n} := \{G = (V, E) \in \mathcal{G} : |V| = n, |E| = m\}$.
- Let

$$G_{m,n}^* \in \arg \max_{G \in \mathcal{G}_{m,n}} \gamma(G). \quad (31)$$

Note that $G_{m,n}^*$ is not necessarily unique for a given m and n .

- For a graph $G = (V, E)$, its *complement* is $\bar{G} = (V, \bar{E})$, where $(i, j) \in E \iff (i, j) \notin \bar{E}$.

Theorem 2: For any integers $m \geq 0$ and $n > 0$ such that $m \leq \frac{1}{2}n(n-1)$,

$$\frac{1}{r^*} \geq \gamma(G_{m,n}^*) \geq \gamma(\hat{T}(n, r^*)) = \frac{1}{r^* + 1} \quad (32)$$

where

$$r^* = \left\lceil \frac{n^2}{2m + n} \right\rceil \quad (33)$$

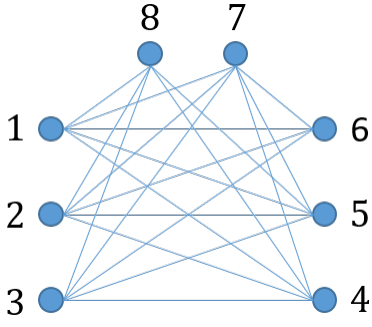
and $\hat{T}(n, r) \in \mathcal{G}$ for any positive integer r is a graph constructed with the following algorithm:

- 1) Partition the vertices into r different sets $C_1, \dots, C_r$ such that $|C_i|$ and $|C_j|$ differ by no more than 1 for all $i, j \in \{1, \dots, r\}$. In other words, all sets in the partition are as close to equal size as possible.
- 2) Create edges between all nodes within each set.

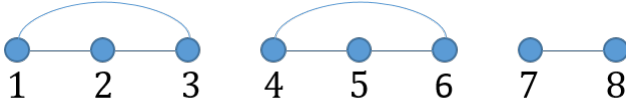
Prior to the proof, we give some insight into the implication of this theorem. With some algebraic manipulation, (32) becomes

$$1 \geq \frac{\gamma(\hat{T}(n, r^*))}{\gamma(G_{m,n}^*)} \geq \frac{r^*}{r^* + 1}. \quad (34)$$

This format shows that the worst-case efficiency of $G_{m,n}^*$ can be approximated by that of $\hat{T}(n, r^*)$. The approximation gets closer as r^* increases, which according to (33) corresponds to when m is on the order of n and not n^2 . In other words, r^*



(a) The Turán graph $T(8, 3)$, where the clique number is 3. No other graph with 8 vertices can have more edges without also having a clique of size 4 or higher.



(b) The complement Turán graph $\hat{T}(8, 3)$, where the independence number is 3. No other graph with 8 vertices can have less edges without also having an independent set of size 4 or higher.

Fig. 6: A Turán graph and its complement

is high for sparse graphs. This idea is represented in Figure 5.

This result gives insight into how to construct a graph given n vertices and m edges: construct the graph $\hat{T}(n, r^*)$. As an example, if a system designer had 8 agents but could only place 7 edges, an optimal structure would be that in Figure 6b. Implied in the following proof is that in the case where n is divisible by r^* , $G_{m,n}^* = \hat{T}(n, r^*)$. However, when this is not true, $\hat{T}(n, r^*)$ is an approximation of $G_{m,n}^*$, since $G_{m,n}^*$ has more edges. Theorem 1 does not give insight as to where to place these extra edges. Intuitively, this leads to the slack between the upper and lower bounds in Theorem 2, although technically it need not be the case that $\hat{T}(n, r^*)$ is an induced subgraph of $G_{m,n}^*$. We now proceed with the proof.

Proof: We first describe some properties of $\hat{T}(n, r)$ as it is constructed in the theorem statement. 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 different sets $C_1, \dots, C_r$ such that $|C_i|$ and $|C_j|$ differ by no more than 1 for all $i, j \in \{1, \dots, r\}$.
- 2) Create edges between all nodes not within the same set.

The number of edges $\bar{m}$ in $T(n, r)$ satisfies the following inequality

$$\bar{m} \leq \left(1 - \frac{1}{r}\right) \cdot \frac{n^2}{2}. \quad (35)$$

A result known as Turán's theorem states that $T(n, r)$ is an n -vertex graph with the most edges that has clique number

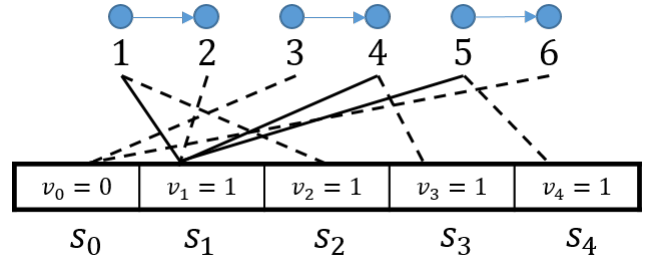


Fig. 7: A graph $\hat{T}(6, 3)$ with a weighted set cover problem described in the proof for Theorem 2, where we know $k(\hat{T}(6, 3)) = 3$. In this instance, $y = 1$, $z = 2$ and the set $I = \{2, 4, 6\}$. By the dotted lines we see that the optimal choices yield $f(x_{1:6}^*) = 4$ and in the worst-case, the generalized distributed greedy algorithm yields $f(x_{1:6}) = 1$. Therefore $\gamma(f, X, \hat{T}(6, 3)) = \frac{1}{4}$, and the lower bound on $\gamma(\hat{T}(6, 3))$ given in Theorem 1 is tight.

r or smaller [21]. Alternatively stated,

$$T(n, r) \in \arg \max_{G=(V,E): \omega(G) \leq r} |E|. \quad (36)$$

Note that Step 1 above is the same as in the procedure for creating $\hat{T}(n, r)$, but Step 2 is different. Any edge in $T(n, r)$ will not be present in $\hat{T}(n, r)$, and vice versa, thus these graphs are complements of each other, i.e.,

$$\hat{T}(n, r) = \overline{T(n, r)}. \quad (37)$$

It is also well-known in graph theory that an independent set of vertices in a graph G forms a clique in $\bar{G}$. This implies that $\alpha(G) = \omega(\bar{G})$. Combining this fact with (37) and (36), we conclude that

$$\hat{T}(n, r) \in \arg \min_{G=(V,E): \alpha(G) \leq r} |E|. \quad (38)$$

In words $\hat{T}(n, r)$ is a graph with the least edges that has independence number r or fewer. An example of a Turán graph and its complement is found in Figure 6.

We now show the rightmost equality in (32). By construction $\hat{T}(n, r)$ is a series of r disconnected cliques $C_1, \dots, C_r$, therefore $k(\hat{T}(n, r)) = r$. By Theorem 1 it follows that $\gamma(\hat{T}(n, r)) \geq \frac{1}{r+1}$. However, to show that this bound is tight, we again leverage the weighted set cover problem. Let y and z be the first and second vertices, respectively, in some clique. Let $s_0, \dots, s_{r+1}$ be the partition on S , where $v_0 = 0, v_1 = \dots = v_{r+1} = 1$, and let I be a set of nodes, one from each clique C_j , that includes z . Then the X_i are assigned as follows:

$$X_i = \begin{cases} \{s_1\} & \text{if } i = z \\ \{s_1, P(i)\} & \text{if } i \in I \\ \{s_0\} & \text{otherwise,} \end{cases}$$

where $P : I \rightarrow \{s_2, \dots, s_{r+1}\}$ is injective. An instance of such a weighted set cover problem is shown in Figure 7. Here the optimal sequence is for each $i \in I$ to choose $P(i)$, yielding $f(x_{1:n}^*) = r + 1$. However, the worst-case generalized distributed greedy algorithm set of decisions is

when all $i \in I$ choose s_1 , so that $f(x_{1:n}) = 1$. Therefore, $\gamma(f, X, \hat{T}(n, r)) = \frac{1}{r+1}$. Since this example meets the lower bound shown above, it follows that $\gamma(\hat{T}(n, r)) = \frac{1}{r+1}$. The rightmost equality in (32) is the case where $r = r^*$.

Next we show the middle inequality in 32: $\gamma(G_{m,n}^*) \geq \gamma(\hat{T}(n, r^*))$. We claim that

$$r^* = \min_{\hat{T}(n,r): \hat{m} \leq m} r, \quad (39)$$

where $\hat{m}$ is the number of edges in $\hat{T}(n, r)$. The equations (37) and (35) imply that in order to guarantee $\hat{T}(n, r)$ does not have more than m edges the following must be true:

$$\frac{1}{2}n(n-1) - m \leq \left(1 - \frac{1}{r}\right) \frac{n^2}{2}, \quad (40)$$

since $\bar{m} = \frac{1}{2}n(n-1) - m$. With some algebraic manipulation, this implies

$$\frac{n^2}{2m+n} \leq r. \quad (41)$$

Since r must be a positive integer, the lowest value of r is r^* as defined in (33), therefore (39) holds.

Let $G_{m,n} \in \mathcal{G}_{m,n}$ be a graph which is created by starting with $\hat{T}(n, r^*)$ and adding $m - \hat{m}$ edges. Since adding edges cannot remove any cliques, $k(G_{m,n}) \leq k(\hat{T}(n, r^*)) = r^*$. Thus by Theorem 1

$$\gamma(G_{m,n}^*) \geq \gamma(G_{m,n}) \geq \frac{1}{k(G_{m,n})+1} \geq \frac{1}{r^*+1}, \quad (42)$$

proving the middle inequality.

Finally, we show the leftmost inequality $\frac{1}{r^*} \geq \gamma(G_{m,n}^*)$. According to (38) and (39), one cannot achieve a lower independence number than r^* in a graph with n vertices and m edges. Therefore, by Theorem 1 this inequality holds. ■

V. CONCLUSION

In this paper we have shown bounds on the worst-case efficiency of the distributed greedy algorithm for submodular maximization. These bounds show how to design communication structures that maximize the worst-case efficiency.

Future research can follow in several directions. For example, while the bounds presented in Section III are applicable to any graph structure $G \in \mathcal{G}$, it is not clear how to characterize graphs where $\gamma(G) = \frac{1}{\alpha(G)}$ versus those where $\gamma(G) = \frac{1}{k(G)+1}$. It is also not clear whether there exists any graph such that $\frac{1}{\alpha(G)} > \gamma(G) > \frac{1}{k(G)+1}$. Precisely characterizing $\gamma(G)$ for any graph structure will also lead to an exact formulation on how to construct any $G_{m,n}^*$.

Another idea for future research is to consider a game-theoretic approach instead of a greedy approach, similar to work done in [22]. In this case, the order of the agents wouldn't matter, and the goal would be to define characteristics of 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. The key point of study here would be whether a different set of functions could yield better performance guarantees than marginal contribution.

REFERENCES

- [1] Olga Barinova, Victor Lempitsky, and Pushmeet Kohli. On detection of multiple object instances using hough transforms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1773–1784, 2012.
- [2] Niv Buchbinder, Moran Feldman, Joseph Seffi, and Roy Schwartz. A tight linear time (1/2)-approximation for unconstrained submodular maximization. *SIAM Journal on Computing*, 44(5):1384–1402, 2015.
- [3] Marshall L Fisher, George L Nemhauser, and Laurence A Wolsey. An analysis of approximations for maximizing submodular set functions II. In *Polyhedral combinatorics*, pages 73–87. Springer, 1978.
- [4] Bahman Ghahesifard and Stephen L Smith. On distributed submodular maximization with limited information. In *American Control Conference (ACC)*, pages 1048–1053. IEEE, 2016.
- [5] Martin Grötschel, László Lovász, and Alexander Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981.
- [6] Satoru Iwata, Lisa Fleischer, and Satoru Fujishige. A combinatorial strongly polynomial algorithm for minimizing submodular functions. *Journal of the ACM (JACM)*, 48(4):761–777, 2001.
- [7] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM, 2003.
- [8] Pushmeet Kohli, M Pawan Kumar, and Philip HS Torr. P³ & beyond: Move making algorithms for solving higher order functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(9):1645–1656, 2009.
- [9] Andreas Krause and Carlos Guestrin. Near-optimal observation selection using submodular functions. In *AAAI*, volume 7, pages 1650–1654, 2007.
- [10] Andreas Krause, Ram Rajagopal, Anupam Gupta, and Carlos Guestrin. Simultaneous placement and scheduling of sensors. In *Proceedings of the 2009 international conference on information processing in Sensor Networks*, pages 181–192. IEEE Computer Society, 2009.
- [11] Hui Lin and Jeff Bilmes. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 510–520. Association for Computational Linguistics, 2011.
- [12] László Lovász. Submodular functions and convexity. In *Mathematical Programming The State of the Art*, pages 235–257. Springer, 1983.
- [13] Jason R Marden. The role of information in multiagent coordination. In *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, pages 445–450. IEEE, 2014.
- [14] Jason R. Marden. The role of information in distributed resource allocation. *IEEE Transactions on Control of Network Systems*, 2016.
- [15] Michel Minoux. Accelerated greedy algorithms for maximizing submodular set functions. In *Optimization Techniques*, pages 234–243. Springer, 1978.
- [16] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions I. *Mathematical Programming*, 14(1):265–294, 1978.
- [17] Guannan Qu, Dave Brown, and Na Li. Distributed greedy algorithm for multi-agent task assignment problem with submodular utility functions. Preprint submitted to Automatica, December 2016.
- [18] Alexander Schrijver. A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *Journal of Combinatorial Theory, Series B*, 80(2):346–355, 2000.
- [19] Amarjeet Singh, Andreas Krause, Carlos Guestrin, William J Kaiser, and Maxim A Batalin. Efficient planning of informative paths for multiple robots. In *IJCAI*, volume 7, pages 2204–2211, 2007.
- [20] Maxim Sviridenko. A note on maximizing a submodular set function subject to a knapsack constraint. *Operations Research Letters*, 32(1):41–43, 2004.
- [21] Paul Turán. On an extremal problem in graph theory. *Mat. Fiz. Lapok*, 48(436–452):137, 1941.
- [22] Adrian Vetta. Nash equilibria in competitive societies, with applications to facility location, traffic routing and auctions. In *Foundations of Computer Science, 2002. Proceedings. The 43rd Annual IEEE Symposium on*, pages 416–425. IEEE, 2002.
- [23] Jan Vondrák. Optimal approximation for the submodular welfare problem in the value oracle model. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 67–74. ACM, 2008.