Strategic Information Sharing in Greedy Submodular Maximization

David Grimsman, João P. Hespanha and Jason R. Marden

***Abstract*— Submodular maximization is an important problem with many applications in engineering, computer science, economics and social sciences. Since the problem is NP-Hard, a greedy algorithm has been developed, which gives an approximation within 1/2 of the optimal solution. This algorithm can be distributed among agents, each making local decisions and sharing that decision with other agents. Recent work has explored how the performance of the distributed algorithm is affected by a degradation in this information sharing. This work introduces the idea of strategy in these networks of agents and shows the value of such an approach in terms of the performance guarantees that it provides. In addition, an optimal strategy that gives such guarantees is identified.**

I. Introduction

Submodular optimization is a well-studied topic due to its wide applicability in engineering, computer science, economics, and social problems. Examples include information gathering [12], the spread of influence in a social network [10], image processing [11], document summarization [14], path planning for multiple robots [20], sensor placement [13], and resource allocation in multi-agent systems [16]. The key thread in these problems is that each exhibits some form of a "diminishing returns" property, e.g., adding more sensors to a sensor placement problem improves performance, but the rate of improvement degrades as more sensors get added to the system. Any problem exhibiting such behavior can likely be formulated as a submodular optimization problem.

While submodular minimization can be solved in strongly polynomial time using a convex relaxation [19], [9], submodular maximization has been shown to be NP-Hard in certain cases [15]. Therefore, much research has focused on developing fast algorithms to attain near-optimal solutions to such maximization problems. A resounding message from this research is that there are often simple algorithms that give strong guarantees on the quality of the solution associated with submodular maximization problems [1].

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

This research was supported by ONR grant #N00014-17-1-2060 and NSF Grant #ECCS-1638214.

The seminal work in [6] 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 pushes these guarantees from $1/2$ to $1-1/e$ [17], [2], [5]. Further, it is important to highlight that no polynomial-time algorithm can achieve a higher guarantee than $(1-1/e)$, unless $P = NP$ [4].

The distributed nature of the greedy algorithm is also appealing from the standpoint of distributed control. In the greedy algorithm, each agent sequentially makes a decision using information by optimizing the global objective conditioned on the decisions taken by the preceding agents. While simplistic in nature, the greedy algorithm requires that each agent has access to the decisions of all preceding agents. In many cases this information exchange is costly or infeasible.

A recent string of research has sought to characterize how reducing this informational burden on the agents impacts the performance of the greedy algorithm for submodular maximization problems [7], [8]. The two take-away messages from this work are the following: (i) reducing the information available to the agents negatively impacts the efficiency guarantees associated with the greedy algorithms; and (ii) not all information is equally valuable in terms of efficiency guarantees. The second point is particularly interesting and demonstrates the need to optimize over what to share when there are constraints on the amount of information that can propagate throughout the system.

In light of these findings, this paper explores how *strategic* information exchange and selection can be leveraged to compensate for the loss in performance associated with greedy algorithm subject to informational restrictions.

– *Strategic information sharing:* In the nominal greedy algorithm, each agent is aware of the decision of all preceding agents. In a more localized setting where agents can only communicate with a given subset of preceding agents, a natural starting point is the case where agents are informed of the decision of these preceding agents [7], [8]. Alternatively, one could imagine a setting where

agents strategically communicate information regarding their decision or other decisions that they are aware of. Can such strategic information exchange help improve the efficiency guarantees of the greedy algorithm with information restrictions?

– *Strategic selection:* In the nominal greedy algorithm, each agent chooses its decision by optimizing a global objective subject to the information the agent can access. Could alternative selection rules, e.g., optimize a perturbed version of the global object subject to available information, yield more desirable efficiency guarantees?

The efficiency guarantees associated with a strategic greedy algorithm subject to informational limitations depends on both the specification of the strategic information exchange and the strategic processing. The main findings of this paper characterize these strategic policies that optimize the efficiency guarantees associated with the strategic greedy algorithm subject to informational limitations. As we demonstrate in Theorems 2 and 3, strategic information exchange can offer significant improvements over non-strategic information exchanges without imposing greater communication demands.

II. Model

A. Submodularity and the greedy algorithm

Let S be a finite set of elements and consider any function $f : 2^S \to \mathbb{R}_{\geq 0}$ that satisfies the following properties:

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

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

We will generally refer to such functions as merely submodular functions. Submodular functions have been used to model several problems relevant to engineering systems: estimation with limited measurements [21] and task allocation [3], [18], among many others. Consequently, there has been significant research efforts geared at deriving algorithms for optimizing a given submodular objective function subject to a set of constraints on the underlying decision variables. More formally, for a set of decision variables $\mathcal{X} \subseteq 2^S$, the goal of such an algorithm is to derive the decision variable that optimizes

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

It is widely known that the optimization problem in (2) is NP-Hard. Nonetheless, researchers have had significant success deriving algorithms that attain near optimal solutions for various classes of submodular objective functions and constraints on the decision variables [17], [2].

In this paper we focus on distributed solutions for attaining near optimal solutions to the optimization problem given in (2). Here, there are a collection of decision-making entities, which we will henceforth refer to as agents, denoted by $N = \{1, ..., n\}$ and each agent $i \in N$ is associated with an action set $X_i \subseteq 2^S$. We will denote a collective decision by the tuple $x = (x_1, \ldots, x_n) \in X = X_1 \times \cdots \times X_n$, and will express the quality of a collective decision x as $f(x)$ with the understanding that $f(x) = f(x_1 \cup \cdots \cup x_n)$.

One of the most well-studied distributed algorithms for computing a near-optimal solution is the greedy algorithm [6]. The greedy algorithm to (2) proceeds as follows: each agent $i \in N$ sequentially makes its choice x_i^{sol} according to the rule

$$x_i^{\text{sol}} \in \arg\max_{x_i \in X_i} f(x_i, x_1^{\text{sol}}, \ldots, x_{i-1}^{\text{sol}}), \quad (3)$$

i.e., each agent i selects the action that would optimize the objective f given knowledge of the action choices of the prior agents. One of the notable results associated with the greedy algorithm ensures that the emergent solution will be at least 50% efficient when compared to the optimal solution, i.e.,

$$\frac{f(x_1^{\text{sol}}, \ldots, x_n^{\text{sol}})}{f(x^{\text{opt}})} \geq \frac{1}{2}, \quad (4)$$

where x^{opt} is defined in (2). It is important to highlight that these 50% guarantees hold for any submodular function f and agents' action sets $X_1, \ldots, X_n$.

B. An informationsl greedy algorithm

While simplistic in nature, the greedy algorithm imposes two structural considerations on the agents' decision-making rules:

1) Each agent $i \in N$ is required to have knowledge of the decisions of previous agents, which could be gathered through either sensing or communication.
2) Each agent leverages the rule in (5) to select a decision, as opposed to any other method.

We can relax the first constraint by allowing that each such agent $i \in N$ only has knowledge of the action choice of a set of agents $\mathcal{N}_i \subseteq \{1, \ldots, i-1\}$ and the

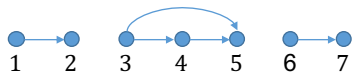


Fig. 1: An example graph structure showcasing limited information in the greedy algorithm. For instance, agent 5 can only utilize the decisions of agents 3 and 4, so $\mathcal{N}_5 = \{3, 4\}$ and there are edges $(3, 5)$ and $(4, 5)$. This graph G also provide the strongest performance given 7 nodes and 5 edges, where $\gamma(G) = 1/4$. This means that no other graph G with 7 nodes and 5 edges can have a higher efficiency. This is indicative of the results in [8] that building disconnected cliques close in size best utilizes the edges in the greedy algorithm.

extension of of the selection rule in (3) is now of the form

$$x_i^{\text{sol}} \in \arg\max_{x_i \in X_i} f\left(x_i, \{x_j^{\text{sol}}\}_{j \in \mathcal{N}_i}\right). \tag{5}$$

The findings presented in [7] demonstrate that the efficiency guarantees associated with this informationally restrictive greedy algorithm can be interpreted by analyzing an acyclic information graph $G = (N, E)$ where the agent set is the set of nodes, and a directed edge $(i, j) \in E$ if and only if $j \in \mathcal{N}_i$ (see Figure 1). We denote the set of such information graphs by $\mathcal{G}$. The solution of this local greedy algorithm is referred to as $x^{\text{sol}}(f, X, G)$ and the optimal decision as $x^{\text{opt}}(f, X)$ to explicitly highlight the dependence. We will denote the efficiency guarantees associated with a graph G as

$$\gamma(G) = \inf_{f,X} \frac{f\left(x^{\text{sol}}\left(f, X, G\right)\right)}{f\left(x^{\text{opt}}\left(f, X\right)\right)}. \tag{6}$$

Note that $x^{\text{sol}}(f, X, G)$ could in fact be a set when (5) is not unique, so we write $f\left(x^{\text{sol}}\left(f, X, G\right)\right)$ with the understanding that f is evaluated at the worst possible candidate solution, i.e., $\min_{x \in x^{\text{sol}}(f,X,G)} f(x)$.

The findings in [8] can be summarized as follows:

Theorem 1 (Grimsman et al., 2017 [8]): Let $G \in \mathcal{G}$ be any information graph. Then

$$\frac{1}{\alpha(G)} \geq \gamma(G) \geq \frac{1}{1 + k(G)}, \tag{7}$$

where $\alpha(G)$ is the independence number and $k(G)$ is the clique cover number of the graph G.

An interesting consequence of the characterization presented in Theorem 1 is that it provides insight into the graph structures that have the highest efficiency given a constraint on the number of edges in the graph (see Theorem 2, [8]). Informally, the optimal graph structures involve allocating the edges to form the minimal number of disjoint cliques. See Figure 1 for an illustration. Recall that a clique formed over a set of agents $C \subseteq N$ implies that if $i, j \in C$ such that $j < i$, then $(i, j) \in E$.

In light of these results, this work seeks to address the question of whether strategic information exchange and decision-making can help compensate for the efficiency loss associated with localizing information in the greedy algorithm. Thus we focus our attention on the optimal graph structures highlighted above where the agents are partitioned into a series of disconnected cliques $\mathcal{C} = \{C_1, ..., C_m\}$. We denote $C_i > C_j$ to mean that the agents of C_i come before the agents in C_j in the sequence. As an example, for the graph in Figure 1, $C_1 = \{1, 2\}, C_2 = \{3, 4, 5\}, C_3 = \{6, 7\}$. We will henceforth refer to a graph as $\mathcal{C}$ when appropriate to avoid confusion. For such graph structures, [8] demonstrates that the resulting efficiency guarantees associated with the localized greedy algorithm are precisely

$$\gamma(\mathcal{C}) = \frac{1}{1 + |\mathcal{C}|}, \tag{8}$$

as $k(\mathcal{C}) = \alpha(\mathcal{C}) = |\mathcal{C}|$, and these graphs hit the lower bound in Theorem 1.

C. Strategy rules

In this work, we assume that there is limited information sharing among the cliques of agents in $\mathcal{C}$. Clique C shares information z_C with agents in future cliques using the following rule

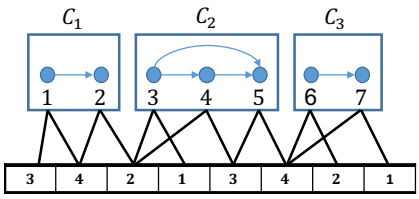
$$z_C = I_C(f, X_C, \{z_{C'}\}_{C' < C}) \in \{x_i^{\text{sol}}\}_{i \in C}, \tag{9}$$

where $X_C = \bigcup_{i \in C} X_i$. We refer to I_c as the *information strategy* for clique C and $I = \{I_C\}_{C \in \mathcal{C}}$ as the information strategy profile for $\mathcal{C}$. One thing to note about (9) is that it imposes a restriction on the amount of information that can be shared by C: it can only share the decision of a single agent. The information strategy I_c instructs C on which agent.

This work also focuses on the selection rule for each agent. The literature is clear that in the presence of full information, a selection rule that has the highest guarantees is (3). However, this notion has not been addressed in the case when agent i receives limited information, nor in the presence of information strategy. Therefore, rather than assuming agent i chooses according to (5), we use a more general approach and let $i \in C$ choose according to the following rule:

$$x_i^{\text{sol}} = \pi_{i,C}(f, \{x_i\}_{i \in \mathcal{N}_i}, \{z_{C'}\}_{C' < C}). \tag{10}$$

We refer to $\pi_{i,C}$ as the *selection strategy* for agent i in clique C and $\pi = \{\pi_{i,C}\}_{i \in N}$ as the selection strategy



$\{I,\pi\}$	Agent 1 choice	Agent 2 choice	Agent 3 choice	Agent 4 choice	Agent 5 choice	Agent 6 choice	Agent 7 choice	z_{C_1}	z_{C_2}	Value
Optimal	L	L	R	L	L	R	L	-	-	19
$\{I^P,\pi^M\}$	R	R	L	R	R	L	L	x_1^{sol}	x_3^{sol}	13
$\{I^M,\pi^M\}$	R	R	L	R	R	R	R	x_1^{sol}	x_5^{sol}	16

Fig. 2: An example scenario for the greedy algorithm including strategy. The disconnected boxes along the bottom represent S, and f is simply the sum of the values in a set of boxes. The blue circles are the agents, and the black lines represent X: each agent can choose between two boxes. In the table, these choices are denoted L for left and R for right. The partition $\mathcal{C}$ has 3 sets: C_1, C_2, C_3. Each set forms a clique among the agents (seen with the blue edges) in its set. Results for two strategies are shown in the table.

profile of $\mathcal{C}$. It should be clear that (5) is a special case of (10), but other selection strategies are possible. When all $\pi_{i,C}$ use (5), we denote this as π^M, since agents choose the action that delivers the highest marginal contribution. Note that the information strategy and the selection strategy influence each other: I_c requires knowledge of x_i^{sol} for $i \in C$ and $\pi_{i,C}$ requires knowledge of $z_{C'}$ for all $C' < C$.

D. Examples

We now give two examples of information strategies and show how they are utilized in the greedy algorithm. The first is a pre-committed strategy:

$$z_C = x_i^{\text{sol}}, \tag{11}$$

where $i \in C$ is determined a priori. The second is

$$z_C = \arg\max_{i \in C} \Delta(x_i^{\text{sol}} | \{x_j^{\text{sol}}\}_{j \in \mathcal{N}_i}, \{z_{C'}\}_{C' < C}), \tag{12}$$

where for $A, B \subseteq S$, $\Delta(A|B) := f(AB) - f(B)$. Essentially, this strategy shares the decision of the agent with the highest margianl contribution. As we will be referring to these strategies often, denote I^P as the profile when I_C is (11) and I^M as the profile when I_C is (12) for all $C \in \mathcal{C}$. Figure 2 gives an example to show how these strategies work.

E. Efficiency

This new model requires an updated definition for efficiency: one can extend (6) to be

$$\gamma(\mathcal{C}, \{I, \pi\}) = \inf_{f,X} \frac{f(x^{\text{sol}}(f, X, \mathcal{C}, \overbrace{\{I, \pi\}}^{\text{strategic component}}))}{f\left(x^{\text{opt}}\left(f, X\right)\right)}. \tag{13}$$

With this definition, (8) is rewritten as

$$\gamma(\mathcal{C}, \{I, \pi\}) = \frac{1}{1 + |\mathcal{C}|}, \tag{14}$$

where $z_C = \emptyset$ for all C and $\pi = \pi^M$. Given the baseline in (14), we use γ as a metric to gauge how strategic information exchange and selection can help compensate for efficiency loss due to localized information. In relation to this metric, we denote the optimal strategy as $\{I^*, \pi^*\}$

III. The Benefit of Strategy

In this section, we characterize the benefit of using strategies in the greedy algorithm. To begin, we show the efficiency of the greedy algorithm when a pre-committed information strategy is used, which can be considered an application of Theorem 1.

Corollary 1: Let $\mathcal{C}$ be a partition of agents such that $|C| > 1$ for all C, and assume $I = I^P$ and $\pi = \pi^M$. Then

$$\gamma(\mathcal{C}, \{I, \pi\}) = \frac{1}{1 + |\mathcal{C}|}. \tag{15}$$

We omit a formal proof here, but essentially this statement follows from the observation that utilizing I^P and π^M is equivalent to adding edges to the graph $\mathcal{C}$ between the partitions. Since no clique is sharing the information of every agent, the clique cover number remains the same, and the efficiency is the same as in (14).

Comparing Corollary 1 to the baseline in (14), we can glean that if I pre-commits to sharing the decision of a set of agents, then this information sharing strategy offers no benefit to efficiency. By pre-comitting, one is essentially removing the dependence of I on f from (9). Even in the case that we relax the restriction in (9) that only one piece of information is shared, we see no benefit to efficiency, assuming that the decision of at least one agent in every clique is not shared. Therefore, any strategy that increases γ must utilize f.

Theorem 2: Let $\mathcal{C} = \{C_1, .., C_m\}$ be a partition such that $|C_i| > 1$ for all i. Then

$$\gamma(\mathcal{C}, \{I, \pi\}) \leq \frac{1}{2 + \sum_{i=1}^{m-1} \prod_{j=1}^{i} (1 - 1/|C_j|)}, \quad (16)$$

with equality when $I = I^* = I^M$ and $\pi = \pi^* = \pi^M$.

The proof for this theorem will be shown at the end of the section, in favor of some discussion up front. We will make a few observations in order to gain some intuition for the expression in (16), and then we will discuss the optimal strategy $\{I^*, \pi^*\}$ further in Section IV.

First, if $\mathcal{C}$ is comprised of a single clique (i.e. $\mathcal{C} = \{C_1\}$), then the expression simplifies to the familiar 1/2 guarantee. This is also true for the expression in Corollary 1. For the rest of the discussion we assume this is not the case.

Next, any term in the sum in (16) is strictly less than 1. The fact that there are $m - 1$ terms in the sum then confirms that $\gamma(\mathcal{C}, \{I^*, \pi^*\}) > \gamma(\mathcal{C}, \{I^P, \pi^M\})$. This is significant, because as mentioned above it holds even when relaxing the information constraint in (9).

To further this point, if one restricted information sharing to only be between cliques C and C' ($C < C'$) then (16) becomes

$$\gamma(\mathcal{C}) = \frac{1}{(1 - 1/|C|) + |\mathcal{C}|}, \quad (17)$$

which still has a strictly higher performance guarantee than (15).

Another set of observations is derived from how $\{I^*, \pi^*\}$ perform as $m \to \infty$. First, if $|C| = \omega$ for all $C \in \mathcal{C}$, then 16 becomes

$$\gamma(C, \{I^*, \pi^*\}) = \frac{1}{1 + \sum_{c=0}^{m-1} \left(1 - \frac{1}{\omega}\right)^c}. \quad (18)$$

If $m \to \infty$, then the sum in the denominator becomes a geometric series, and simplifies to the following:

$$\gamma(C, \{I^*, \pi^*\}) > \frac{1}{1 + \omega}. \quad (19)$$

This implies that in cases where all cliques are not the same size, if $\omega' = \max_{C \in \mathcal{C}} |C|$, then this lower bound becomes

$$\gamma(C, \{I^*, \pi^*\}) > \frac{1}{1 + \omega'}. \quad (20)$$

Statements (19) and (20) illustrate one of the key benefits to using strategy: for a fixed ω, the efficiency does not become arbitrarily bad as $m \to \infty$. In Corollary 1, $\gamma(\mathcal{C}, \{I^P, \pi^M\}) \to 0$. However, in the above observations, we can see that when the right strategy is applied, even for minimal information sharing, we are always guaranteed to stay within a bound dictated by the size of the largest clique in $\mathcal{C}$. Informally, a little information with the right strategy is better than a lot of information with a poor strategy.

Another insight from Theorem 2 is found in the proof. One way to think of a worst-case scenario is to consider an adversary attempting to design the worst f and X given the graph and strategy. For graphs in $\mathcal{G}$, [8] shows that, in essence, the adversary only needs to "zero-out" some agents by restricting their actions to the empty set. The agents that remain are independent from each other, and thus make decisions independently. These agents have all the power of decision-making, without the benefit of the information from others. However, as will be shown, the adversary cannot use this approach in the presence of strategy when I^* and π^* are employed. In fact the decision-making power must be equally distributed among agents within a clique, so as to minimize the edge's effectiveness. This shows that optimal strategies spread the power of decision-making more equally among the agents.

A. Proof for Theorem 2

First we give an example which shows a universal upper bound on $\gamma(\mathcal{C}, \{I, \pi\})$. Then we show that if $I = I^M$ and $\pi = \pi^M$, there is a lower bound on $\gamma(\mathcal{C}, \{I^M, \pi^M\})$ which matches the upper bound. Thus the bound is tight and we know that $\{I^M, \pi^M\}$ is the optimal strategy.

1) Upper Bound: Here we give an example of S, f, and X to serve as an upper bound on $\gamma(\mathcal{C}, \{I, \pi\})$, and show that it is the exact expression in (16). Assume to begin that I^M and π^M are used.

We introduce some new notation, for convenience:

- Let $f(i) = f(x_i^{\text{sol}})$ and let $f(i^*) = f(x_i^{\text{opt}})$.
- For set of agents J, let $f(J) = f\big(\bigcup_{i \in i} x_i^{\text{sol}}\big)$.
- For $J = \{a, a+1, ..., b\}$, let $f(a : b) = f(J)$.
- Let $Z_C = \bigcup_{C' < C} z_{C'}$, i.e., Z_C is the set of decisions in prior cliques that are shared with the agents in clique C.
- Let the clique that sequentially comes before C be denoted as $C - 1$.

Assume that S is a set boxes, and that f is simply the area of the boxes covered by the choices of the agents. Suppose there is a box u, where $f(u) = 1$, which will be chosen by the agents in the worst case. There are also boxes $v_1, ..., v_k$, and the optimal choices will allow agents to cover all boxes completely.

(a) The set of worst-case greedy choices as described in the upper bound proof for Theorem 2. Notice that u_{C_1} and u_{C_2} are divided up into 2 and 3 parts, respectively, for each agent in the cliques. Only the first agent in C_3 chooses u_{C_3}.

(b) The set of optimal choices as described in the upper bound for the Theorem 2 proof. Notice that agent 7 chooses the full set of boxes chosen in the greedy algorithm.

Fig. 3: An example of the upper bound f, X for Theorem 2. As argued in the proof, any I and π must value every decision and agent, respectively, the same. In this case, $f(x^{\text{sol}}) = 1$ and $f(x^{\text{opt}}) = 2 + 1/2 + 1/2(2/3) = 17/6$, so $\gamma(f, X, \mathcal{C}, \{I, \pi\}) = 6/17$. Theorem 2 also states that this is the max possible efficiency, i.e. $\gamma(\mathcal{C}, \{I^*, \pi^*\}) = 6/17$.

Each clique C will be able to choose between some portion of u, called u_C, and v_C, where $f(u_C) = f(v_C)$. Additionally each u_C and v_C are divided up equally into $|C|$ parts, so that the value of each agent's choice within the clique is the same. We define $u_1 = u$, and u_C is the portion of u_{C-1} not covered by z_{C-1}, thus

$$f(u_C) = \frac{|C-1|-1}{|C-1|} f(u_{C-1}). \tag{21}$$

In the last clique, the first agent can choose between u_m and v_m, and the second agent can only choose u. All other agents in the last clique are ignored. See Figure 3 for an example.

Agents are equally incentivized to choose their portion in u, so $f(1:n) = 1$. As stated, the optimal choices yield the set $\{u, v_1, ..., v_k\}$. Each v_c is such that

$$f(v_C) = \prod_{C' < C} \frac{|C'|-1}{|C'|}, \tag{22}$$

where $v_1 = 1$ by convention. Therefore, if $\mathcal{C} = \{C_1, .., C_M\}$, then $f(1^* : n^*) = f(u) + \sum_C f(v_C) = 2 + \sum_{i=1}^{m-1} \prod_{j=1}^{i} \frac{|C_j|-1}{|C_j|}$.

Although we intially assumed to use $\{I^M, \pi^M\}$, we now make the claim that this canonical example serves as an upper bound on $\gamma(\mathcal{C}, \{I, \pi\})$ for any I and π. From (9), it is clear that information sharing strategy I_C can only leverage information from past cliques and the action sets of agents in C. *Based on this information, all agents in C are equivalent*. Any choice of z_C will yield the same efficiency guarantee (refer again to Figure 3). Therefore $\gamma(\mathcal{C}\{I, \pi\})$ is the same for any I. A similar argument is made for π. We conclude that the upper bound found by this canonical example is an upper bound on the efficiency for any strategy used.

2) Lower Bound: We will make use of this Lemma, previously proven:

Lemma 1: [8] Let $A \subseteq S$ and $B \subseteq V$. Then

$$f(A, x_B) = f(A) + \sum_{i \in B} \Delta(x_i | A, x_{j \in B: j < i}). \tag{23}$$

We assume that $I = I^M$ and $\pi = \pi^M$. Let $\mathcal{C} = \{C_1, ..., C_m\}$, and we will use the notation that $z_{C_k} = z_k$, and likewise for Z_k. For some C_k, let j be the agent whose decision is z_k. Then:

$$\Delta(C_k | Z_k) = \sum_{i \in C_k} \Delta(i | \mathcal{N}_i \cup Z_k), \tag{24}$$

$$\leq |C_k| \Delta(j | \mathcal{N}_j \cup Z_k), \tag{25}$$

where (24) is true by application of Lemma 1 and (25) is true since I^M uses (12) and j is chosen by I^M. Leveraging the definition of Δ, and the fact that $Z_{k+1} = Z_k \cup z_k$, we see that

$$f(Z_{k+1}) \geq \frac{1}{|C_k|} f(C_k, Z_k) + \left(1 - \frac{1}{|C_k|}\right) f(Z_k) \tag{26}$$

For simplicity, let $a_k = 1 - 1/|C_k|$. Then (26) becomes

$$f(Z_{k+1}) \geq (1 - a_k) f(K_k, Z_k) + a_k f(Z_k) \tag{27}$$

Begin with the following inequality (agent i is in clique $C_{k(i)}$):

$$f(1^* : n^*) \leq f(1 : n, 1^* : n^*), \tag{28}$$

$$\leq f(1:n) + \sum_{i=1}^{n} \Delta(i | \mathcal{N}_i \cup Z_{k(i)}), \tag{29}$$

$$\leq f(1:n) + \sum_{k=1}^{m} \Delta(C_k | Z_k), ‘ \tag{30}$$

$$=f(1:n)+\sum_{k=1}^{m} f(C_k,Z_k)-\sum_{k=1}^{m} f(Z_k), \tag{31}$$

$$\leq 2f(1:n)+\sum_{k=1}^{m-1} f(C_k,Z_k)-\sum_{k=1}^{m-1} f(Z_{k+1}), \tag{32}$$

where (28), (29) are true by submodularity, (30) is true by Lemma 1 and our defined π, (31) is true by definition of Δ, and (32) is true by submodularity and definition of Z_{k+1}.

Notice that the two sums have the same number of terms, and we can apply (27) to each term in the second sum and get the following:

$$\begin{aligned} f(1^*:n^*) \leq & (2+a_{m-1})f(1:n) \\ & + \sum_{k=1}^{m-2} a_k f(C_k,Z_k) - \sum_{k=1}^{m-2} a_{k+1} f(Z_{k+1}) \end{aligned} \tag{33}$$

Again we see that both sums have the same number of terms and we apply (27) to get:

$$\begin{aligned} f^*(1:n) \leq & (2+a_{m-2}+a_{m-1}a_{m-2})f(1:n) \\ & + \sum_{k=1}^{m-3} a_k - a_{k+1}(1-a_k) f(C_k,Z_k) \\ & - \sum_{k=1}^{m-3} a_{k+2}a_{k+1} f(Z_{k+1}) \end{aligned} \tag{34}$$

Notice that each application of (27) adds some positive term to the coefficient of $f(1:n)$ and drops a term from each of the sums. Let b_j be the term added to the coefficient of $f(1:n)$ after applying (27) j times. In other words, $b_1 = a_{m-1}$, $b_2 = a_{m-2} + a_{m-1}a_{m-2} - a_{m-1}$, etc. After applying (27) $m-1$ times, we see that

$$f(1^*:n^*) \leq \left(2+\sum_{j=1}^{m-1} b_j\right) f(1:n) \tag{35}$$

Thus to find the lower bound on efficiency, we need to find $\sum_j b_j$. Following the pattern, each b_j can be defined as follows:

$$b_j = \sum_{i=m-j}^{m-1} \prod_{d=m-j}^{i} a_d - \sum_{i=m-j+1}^{k} \prod_{d=m-j+1}^{i} a_d \tag{36}$$

Notice that the second sum for b_j is the negative of the first sum for b_{j-1}. Thus

$$\sum_{j=1}^{m-1} b_j = \sum_{i=1}^{m-1} \prod_{d=1}^{i} a_d. \tag{37}$$

Therefore, the lower bound meets the upper bound.

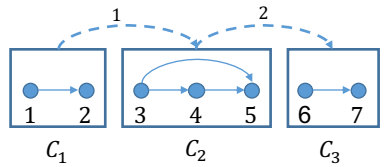


Fig. 4: An exmaple graph H. Notice that $\mathcal{C}$ is still a partition on the agents, and the nodes in H are the sets in the partition. The dotted arrows represent the edges in H: in this case $\mathcal{N}_2 = \{C_1\}$ and $\mathcal{N}_3 = \{C_2\}$. Also, w_1, w_2 are indicated by the numbers above the edges.

IV. Optimal Strategy

In this section, we relax two of the constraints in our original model and show that $\{I^M, \pi^M\}$ is still an optimal strategy. First, we relax the requirement in (9) that only one agent's decision can be passed along. Instead, for clique C_k, we allow $w_C < |C|$ agents' decisions to be used. Additionally, we relax the constraint that every clique shares information with all future cliques. Instead, assume that clique C receives information from $\mathcal{N}_C \subseteq \{C'\}_{C'<C}$. Then we can extend (9):

$$z_C = I_C(f, X_C, \{z_{C'}\}_{C' \in \mathcal{N}_C}) \in \{x_i^{\text{sol}}\}_{i \in C}^{w_C}. \tag{38}$$

One way to model this is again by using a graph $H = (\mathcal{C}, E, W)$, where each clique of agents is a node in the graph, $(C', C) \in E$ implies $C' < C$ and $C' \in \mathcal{N}_C$, and W are the weights on the edges, revealing w_C. Note that even though a weight is associated with an edge, all edges outgoing from C will have the same weight, thus we can index the weights by C. An example of such an H is found in Figure 4.

Using this model, one can build a matrix A, which can be leveraged to find the efficiency. Let $\mathcal{K}$ be the set of all cliques in H of any size. Note the overloading of the term *clique*: the nodes in H are cliques of agents, and here we refer to cliques of nodes in H. Each column in A represents a node in H and each row represents a clique in H. The entries of A are as follows:

$$A_{ij} = \begin{cases} 1 & \text{if } C_j \text{ is the last element in clique } i, \\ w_j/|C_j| & \text{if } C_j \text{ is in clique } i, \text{ but not last,} \\ 0 & \text{otherwise.} \end{cases} \tag{39}$$

Thus it acts as a sort of indicator matrix as to which nodes are in which cliques. For the example in Fig. 4,

$$A = \begin{bmatrix} 1 & 0 & 0 & 1/2 & 0 \\ 0 & 1 & 0 & 1 & 2/3 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix}^T. \tag{40}$$

Theorem 3: For graph H, and strategy $\{I, \pi\}$, the efficiency of the greedy algorithm is less than or equal to

$$\begin{aligned} \max_{y} \quad & y^T \mathbf{1} \\ \text{subject to} \quad & Ay \leq \mathbf{1} \\ & y \geq 0, \end{aligned} \tag{41}$$

with equality when $I = I^* = I^M$ and $\pi = \pi^* = \pi^M$.

In this setting, we omit the proof for this statement, but suffice it to say that the expression in Theorem 2 is the solution to (41) when H is a full clique. We present the result here to show that the strategy $\{I^*, \pi^*\}$ is still best even as fewer agents share with each other, and as agents share more information with others.

One implication of this theorem is that homogeneous agents and cliques of agents lead to the strongest guarantees: there is no benefit to varying decision-making among the agents or strategy among the cliques. A single model can be used to describe the behavior for all. This is useful for modeling and implementation purposes.

The fact that π^* is simply the same rule used in the original work of the greedy algorithm is also convenient. It shows that adding strategy among the cliques does not change what the behavior of the agents should be, so intuition gained in the simpler model can be carried over into this more complex one. The strategy I^* also leverages a similar rule, which values agents that have the highest marginal contribution. Thus one can infer that the most valuable information when using this greedy algorithm is marginal contribution to the group.

V. Conclusion

In this paper, we have extended the model of previous work on the greedy algorithm for submodular maximization to include the idea of ad hoc information sharing and decision-making. The resulting efficiency increase was analyzed and the best strategies for agents and groups of agents were presented. Additionally, we showed that these best strategies hold even in a more general setting.

References

[1] 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.

[2] Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a submodular set function subject to a matroid constraint. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 182–196. Springer, 2007.

[3] Han-Lim Choi, Luc Brunet, and Jonathan P How. Consensus-based decentralized auctions for robust task allocation. *IEEE transactions on robotics*, 25(4):912–926, 2009.

[4] Uriel Feige. A threshold of ln n for approximating set cover. *Journal of the ACM (JACM)*, 45(4):634–652, 1998.

[5] Yuval Filmus and Justin Ward. The power of local search: Maximum coverage over a matroid. In *STACS'12 (29th Symposium on Theoretical Aspects of Computer Science)*, volume 14, pages 601–612. LIPIcs, 2012.

[6] 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.

[7] Bahman Gharesifard and Stephen L Smith. Distributed submodular maximization with limited information. *IEEE Transactions on Control of Network Systems*, 2017.

[8] David Grimsman, Mohd Shabbir Ali, Joao P Hespanha, and Jason R Marden. Impact of information in greedy submodular maximization. In *Decision and Control (CDC), 2017 IEEE 56th Annual Conference on*, pages 2900–2905. IEEE, 2017.

[9] 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.

[10] 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.

[11] Gunhee Kim, Eric P Xing, Li Fei-Fei, and Takeo Kanade. Distributed cosegmentation via submodular optimization on anisotropic diffusion. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 169–176. IEEE, 2011.

[12] Andreas Krause and Carlos Guestrin. Submodularity and its applications in optimized information gathering. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(4):32, 2011.

[13] Andreas Krause, Jure Leskovec, Carlos Guestrin, Jeanne VanBriesen, and Christos Faloutsos. Efficient sensor placement optimization for securing large water distribution networks. *Journal of Water Resources Planning and Management*, 134(6), 2008.

[14] Hui Lin and Jeff Bilmes. Multi-document summarization via budgeted maximization of submodular functions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

[15] László Lovász. Submodular functions and convexity. In *Mathematical Programming The State of the Art*, pages 235–257. Springer, 1983.

[16] Jason R. Marden. The role of information in distributed resource allocation. *Transactions on Control of Network Systems*, 2016.

[17] 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.

[18] Guannan Qu, Dave Brown, and Na Li. Distributed greedy algorithm for mulit-agent task assignment problem with submodular utility functions. Preprint submitted to Automatica, December 2016.

[19] Alexander Schrijver. A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *Journal of Combinatorial Theory, Series B*, 80(2):346–355, 2000.

[20] 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.

[21] John Tisdale, ZuWhan Kim, and J Karl Hedrick. Autonomous uav path planning and estimation. *IEEE Robotics & Automation Magazine*, 16(2), 2009.