Multi-Agent Maximization of a Monotone Submodular Function via Maximum Consensus

Navid Rezazadeh
Mechanical and Aerospace Eng. Dept.
University of California Irvine
Irvine, CA
nrezazad@uci.edu

Solmaz S. Kia
Mechanical and Aerospace Eng. Dept.
University of California Irvine
Irvine, CA
solmaz@uci.edu

Abstract

Constrained submodular set function maximization problems often appear in multi-agent decision-making problems with a discrete feasible set. A prominent example is the problem of multi-agent mobile sensor placement over a discrete domain. However, submodular set function optimization problems are known to be NP-hard. In this paper, we consider a class of submodular optimization problems that consists of maximization of a monotone and submodular set function subject to a uniform matroid constraint over a group of networked agents that communicate over a connected undirected graph. Our objective is to obtain a distributed suboptimal polynomial-time algorithm that enables each agent to obtain its respective policy via local interactions with its neighboring agents. Our solution is a fully distributed gradient-based algorithm using the multilinear extension of the submodular set functions and exploiting a maximum consensus scheme. This algorithm results in a policy set that when the team objective function is evaluated at worst case the objective function value is in \(1 - 1/e - O(1/T)\) of the optimal solution. An example demonstrates our results.

1 Introduction

In recent years, optimal multi-agent sensor placement/dispatch to cover areas for sampling/observation objectives has been of great interest in the robotic community to solve various coverage, exploration, and monitoring problems. Depending on the problem setting, the planning space can be continuous [1–4] or discrete [5–8]. In this paper, our interest is in planning in discrete space. Many optimal discrete policy-making for multi-agent sensor placement problems appear in the form of maximizing a utility for the team [5, 9–13]. A particular subset of these problems that we are interested in is in the form of a set-valued function maximization problem described by

\[
\max_{\mathcal{R} \in \mathcal{I} \subset \mathcal{P}} f(\mathcal{R}),
\]

where \(\mathcal{I} \subseteq 2^{\mathcal{P}}\) is the admissible set of subsets originated from the ground policy set \(\mathcal{P}\), which is the union of the local policy set of the agents; see Fig. 1 for an example.

Problem [1] is known to be NP hard [14]. However, when the objective function is monotone increasing and submodular set function, it is well-known that the sequential greedy algorithm [15] delivers a polynomial-time suboptimal solution with guaranteed optimality bound. For large scale submodular maximization problems, reducing the size of the problem through approximations [16] or using several processing units leads to a faster sequential greedy algorithm, however, with some sacrifices on the optimality bound [17–20].
When the set that the agents can choose their policy from is the same for all agents and the agents are homogeneous, the feasible set \( \mathcal{I} \) of (1) can be spanned by a uniform matroid constraint. For problems with uniform matroid constraint, the sequential greedy algorithm delivers a suboptimal solution whose utility value is no worse than \( 1 - \frac{1}{2} \) times the optimal utility value \([14]\). On the other hand, when the local policy set of the agents is disjoint and/or the agents are heterogeneous, and each agent has to choose one policy from its local set, the feasible set \( \mathcal{I} \) of (1) can be spanned by the so-called partitioned matroid constraint.\(^2\) For problems with partitioned matroid constraint, the sequential greedy algorithm delivers a suboptimal solution whose utility value is no worse than \( \frac{1}{2} \) times the optimal utility value \([15]\). For problems with partitioned matroid constraint, more recently, a suboptimal solution with a better optimality gap is proposed in \([21–24]\) using the multilinear continuous relaxation of a submodular set function. The multilinear continuous relaxation of a submodular set function defined on the vertices of the \( n \)-dimensional hypercube facilitates achieves better optimality bounds for the maximization problem (1) by allowing to use a continuous gradient-based optimization algorithm. That is, the relaxation transforms the main discrete problem into a continuous optimization problem with linear constraints. The optimality bound achieved by the algorithm proposed in \([21]\) is \( 1 - \frac{1}{e} \) bound for a partitioned matroid constraint, which outperforms the sequential greedy algorithm whose optimally bound is \( 1/2 \). However, this solution requires a central authority to find the optimal solution. In sensor placement problems, when the agents are self-organizing autonomous mobile agents with communication and computation capabilities, it is desired to solve the optimization problem (1) in a distributed way, without involving a central authority. A distributed multilinear extension based continuous algorithm that uses an average consensus scheme between the agents to solve (1) subject to partitioned matroid constraint is proposed in \([25]\). However, the proposed algorithm assumes that access to the exact multilinear extension of the utility function is available, whose calculation is exponentially hard.

In this paper, motivated by the improved optimality gap of multilinear continuous relaxation based algorithms, we develop a distributed implementation of the algorithm of \([21]\) over a connected undirected graph to obtain a suboptimal solution for (1) subject to a partitioned matroid constraint. In this algorithm, to manage the computational cost of constructing the multilinear extension of the utility function, we use a sampled based evaluation of the multilinear extension and propose a gradient-based algorithm, which uses a maximum consensus message passing scheme over the communication graph. Our algorithm uses a fully distributed rounding technique to compute the final solution of each agent. Through rigorous analysis, we show that our proposed distributed algorithm achieves, with a known probability, a \( 1 - \frac{1}{e} - O(1/T) \) optimality bound, where \( T \) is the number of times agents communicated over the network. A numerical example demonstrates our results.

The organization of this paper is as follows. In Section 2 we introduce the notation used throughout the paper. Section 3 provides the necessary background on submodularity and submodular functions. We introduce our proposed algorithm in Section 4. A demonstration of our results is provided in Section 5. Appendix A gives the proof of our main results in Section 5. Appendix B gives the auxiliary results that we use in establishing the proof of our main results.

### 2 Notation

We denote the vectors with bold small font. The \( p \)th element of vector \( x \) is denoted by \( x_p \). We denote the inner product of two vectors \( x \) and \( y \) with appropriate dimensions by \( x \cdot y \). Furthermore, we use \( 1 \) as a vector of ones with appropriate dimension. We denote sets with capital curly font. For a \( P = \{1, \cdots, n\} \) and a vector \( x \in \mathbb{R}_{\geq 0}^n \) with \( 0 \leq x_p \leq 1 \), the set \( R_x \) is a random set where \( p \in P \) is in \( R_x \) with the probability \( x_p \). Hence, we call such vector \( x \) as membership probability vector. Furthermore, for \( S \subseteq P \), \( 1_S \in \{0, 1\}^n \) is the vector whose \( p \)th element is 1 if \( p \in S \) and 0 otherwise; we call \( 1_S \) the membership indicator vector of set \( S \). \(|x|\) is the absolute value of \( x \in \mathbb{R} \). By overloading the notation, we also use \(|S|\) as the cordiality of set \( S \). We denote a graph by \( G(A, E) \)

---

\(^1\)see Section 3 for a formal definition of uniform matroid constraint.

\(^2\)see Section 3 for a formal definition of partitioned matroid constraint.
where $A$ is the node set and $E \subset A \times A$ is the edge set. $G$ is undirected if and only $(i,j) \in E$ means that agents $i$ and $j$ can exchange information. An undirected graph is connected if there is a path from each node to every other node in the graph. We denote the set of the neighboring nodes of node $i$ by $N_i = \{ j \in A | (i,j) \in E \}$. We also use $d(G)$ to show the diameter of the graph.

Given a set $F \subset X \times \mathbb{R}$ and an element $(p, \alpha) \in X \times \mathbb{R}$ we define the addition operator $\oplus$ as $F \oplus \{(p, \alpha)\} = \{(u, \gamma) \in X \times \mathbb{R} | (u, \gamma) \in F, u \neq p\} \cup \{(u, \gamma + \alpha) \in X \times \mathbb{R} | (u, \gamma) \in F, u = p\} \cup \{(p, \alpha) \in X \times \mathbb{R} | (p, \gamma) \notin F\}$. Given a collection of sets $\mathcal{F}_j \subset X \times \mathbb{R}$, $j \in \mathcal{N}$, we define the max-operation over these collection as $\text{MAX}_{j \in \mathcal{N}} \mathcal{F}_j = \{(u, \gamma) \in X \times \mathbb{R} | (u, \gamma) \in \mathcal{F} \text{ s.t. } \gamma = \max_{(u, \alpha) \in \mathcal{F}} \alpha\}$, where $\mathcal{F} = \bigcup_{j \in \mathcal{N}} \mathcal{F}_j$.

3 Submodular Functions

A set function $f : 2^P \rightarrow \mathbb{R}_{\geq 0}$ is monotone increasing if for sets $P_1 \subset P_2 \subset P$, $f(P_1) \leq f(P_2)$, holds. Furthermore, defining $\Delta_f(p|S) = f(S \cup \{p\}) - f(S)$, $S \subset P$, the set function $f$ is submoular if for $p \in P \setminus P_2$, $\Delta_f(p|P_1) \geq \Delta_f(p|P_2)$, holds, which shows the diminishing return of a set function. Furthermore, a set function is normalized if $f(\emptyset) = 0$.

In what follows without loss of generality we assume that the ground set $P$ is equal to $\{1, \cdots, n\}$. Submodular functions are functions assigning values to all subsets of a finite set $P$. Using the set membership indicator, equivalently, we can regard them as functions on the boolean hypercube, $f : \{0, 1\}^n \rightarrow \mathbb{R}$. For a submodular function $f : 2^P \rightarrow \mathbb{R}_{\geq 0}$, its multilinear extension $F : [0, 1]^n \rightarrow \mathbb{R}_{\geq 0}$ in the continuous space is

$$
F(x) = \sum_{R \subset P} f(R) \prod_{p \in R} x_p \prod_{p \notin R} (1 - x_p), \quad x \in [0, 1]^n. 
$$

The multilinear extension $F$ of set function $f$ is a unique multilinear function agreeing with $f$ on the vertices of the hypercube. The multilinear extension function has the following properties.

Lemma 3.1 (See [21]) If $f$ is non-decreasing, then $\partial F \partial x_p \geq 0$ for all $p \in P$, everywhere in $[0, 1]^n$ (monotonicity of $F$). If $f$ is submodular, then $\partial^2 F \partial x_p \partial x_q \leq 0$ for all $p, q \in P$, everywhere in $[0, 1]^n$.

An alternative way to perceive $F$ is to observe that it is indeed equivalent to

$$
F(x) = \mathbb{E}[f(\mathcal{R}_x)],
$$

where $\mathcal{R}_x \subset P$ is a set where each element $p \in \mathcal{R}_x$ appears independently with the probabilities $x_p$. Then, it can be shown that taking the derivatives of $F(x)$ yields

$$
\frac{\partial F}{\partial x_p}(x) = \mathbb{E}[f(\mathcal{R}_x \cup \{p\}) - f(\mathcal{R}_x \setminus \{p\})],
$$

and

$$
\frac{\partial^2 F}{\partial x_p \partial x_q}(x) = \mathbb{E}[f(\mathcal{R}_x \cup \{p, q\}) - f(\mathcal{R}_x \cup \{q\} \setminus \{p\}) - f(\mathcal{R}_x \cup \{p\} \setminus \{q\}) + f(\mathcal{R}_x \setminus \{p, q\})].
$$

Constructing $F(x)$ in (3) requires the knowledge of $f(\mathcal{R})$ for all $\mathcal{R} \subset P$, which becomes computationally intractable when the size of ground set $P$ increases. However, with the stochastic interpretation (4) of the multilinear extension and its derivatives, if enough samples are drawn according to $x$, we can obtain an estimate of $F(x)$ with a reasonable computational cost. We can use Chernoff-Hoeffding’s inequality to quantify the quality of this estimation given the number of samples.
Let the policy set of each mobile sensor \( i \in A \) be \( P_i = \{(i, p)|p \in B_i\} \), where \( B_i \subset B \) is the set of the allowable sensor placement points for agent \( i \in A \) out of all the sensor placement points \( B \). Note that by definition, for any two agent \( i, j \in A \), \( P_i \cap P_j = \emptyset \). The sensors are heterogeneous, in the sense that the size of their sensing zone is different. The objective is to place the sensors in points in \( B \) such that the total number of the observed points of interest is maximized. The utility function, the sum of observed points, is known to be a monotone and increasing submodular function of the agent’s sensing zone \([27]\). This sensor placement problem can be formalized as the optimization problem (8).

![Diagram](image)

**Figure 1:** Let the policy set of each mobile sensor \( i \in A \) be \( P_i = \{(i, p)|p \in B_i\} \), where \( B_i \subset B \) is the set of the allowable sensor placement points for agent \( i \in A \) out of all the sensor placement points \( B \). Note that by definition, for any two agent \( i, j \in A \), \( P_i \cap P_j = \emptyset \). The sensors are heterogeneous, in the sense that the size of their sensing zone is different. The objective is to place the sensors in points in \( B \) such that the total number of the observed points of interest is maximized. The utility function, the sum of observed points, is known to be a monotone and increasing submodular function of the agent’s sensing zone \([27]\). This sensor placement problem can be formalized as the optimization problem (8). The agents are communicating over a connected undirected graph and their objective is to obtain their respective placement points by interacting only with their communicating neighbors.

**Theorem 3.1 (Chernoff-Hoeffding’s inequality \([26]\))** Consider a set of \( K \) independent random variables \( X_1, ..., X_K \) where \( a < X_i < b \). Letting \( S_K = \sum_{i=1}^{K} X_i \), then

\[
P[|S_K - E[S_K]| > \delta] < 2e^{-\frac{2\delta^2}{K(b-a)^2}}.
\]

Problems such as sensor placement with a limited number of sensors and ample places of possible sensor placement can be formulated as a set value optimization (1) where the feasible set constraint \( I \) is in the form of matroid. A matroid is defined as the pair \( M = \{P, I\} \) with \( I \subset 2^P \) such that

- \( B \in I \) and \( A \subset B \) then \( A \in I \)
- \( A, B \in I \) and \( |B| > |A| \), then there exists \( x \in B \setminus A \) such that \( A \cup x \in I \)

One of the known matroids is uniform matroid \( I = \{R \subset P, |R| \leq k\} \). A sensor placement scenario where there are \( k \) sensors to be placed in locations selected out of possible locations \( P \) can be described by a uniform matroid. On the other hand, when there is a set of agents \( A \), each with multiple sensors \( k_i, i \in A \), that they can place in a set of specific and non-overlapping part \( P_i \) of a discrete space, this constrained can be described by the partitioned matroid \( M = \{P, I\} \), where \( I = \{R \subset P, |R \cap P_i| \leq k_i\} \) with \( \bigcup_{i=1}^{N} P_i = P \) and \( P_i \cap P_j = \emptyset, i \neq j \).

A matroid polytop is a convex hull defined as

\[
P(M) = \{x \in \mathbb{R}_{\geq 0}^{|P|}, \forall Q \in P, \sum_{p \in Q} x_p \leq r_M(Q)\}
\]

with \( r_M \) to be the rank function of the matroid defined as

\[
r_M(Q) = \max\{|S|, S \subset Q, S \in I\}.
\]

For the partition matroid with \( k_i = 1 \) which is the focus of this paper, the convex hull is such that for \( x = [x_1^T, ..., x_N^T]^T \) with \( x_i \in \mathbb{R}_{\geq 0}^{|P_i|} \) associated with membership probability of policies in \( P_i \), we have \( 1^T x_i \leq 1 \), i.e.,

\[
P(M) = \{x \in \mathbb{R}_{\geq 0}^{|P|}, \sum_{p=1}^{|P_i|} x_{ip} \leq 1, \forall i \in A\}.
\]
Algorithm 1 Practical implementation of the continuous greedy process [21].

1: $x \leftarrow 0$, Init: $t \leftarrow 1$,
2: while $t \leq T$ do
3: Draw $K$ samples of $R$ from $P$ according to $x$
4: for $p \in P$ do
5: Estimate $w_p \sim \mathbb{E}[f(R_x \cup \{p\}) - f(R_x \setminus \{p\})]$
6: end for
7: Solve for $R^* = \arg\max_{R \in I} \sum_{p \in R} w_p$.
8: Update membership vector as $x \leftarrow x + \frac{1}{T} R^*$.
9: $t \leftarrow t + 1$
10: end while
11: Use Pipage rounding to convert the fractional solution $x$ to an integral solution.

Note: $1_{R^*}$ is the $v$ in [10].

4 Problem Statement

We consider a group of $A = \{1, \ldots, N\}$ with communication and computation capabilities interacting over a connected undirected graph $G(A, \mathcal{E})$. These agents aim to solve in a distributed manner the optimization problem

$$\max_{R \in I} f(R) \quad \text{s.t.} \quad I = \{R \subset P \mid |R \cap P_i| \leq 1, i \in A\},$$

where utility function $f : 2^P \to \mathbb{R}_{\geq 0}$ is monotone increasing and submodular set function over the discrete policy space $P = \bigcup_{i=1}^N P_i$, with $P_i$ being the policy space of agent $i \in A$, which is only known to agent $i$. In this paper, we work in the value oracle model where the only access to the utility function is through a black box returning $f(R)$ for a given set $R$. Every agent can obtain the value of the utility function at any subset $R \subseteq P$. The constraint (8b) is the partitioned matroid $M = \{P, I\}$, which ensures that only one policy per agent is selected from each local policy set $P_i$, $i \in A$. An example application scenario of our problem of interest is shown in Fig. 1. Without loss of generality, to simplify notation, we assume that the policy space is $P = \{1, \ldots, n\}$, and is sorted agent-wise with $1 \in P_1$ and $n \in P_N$.

Finding the optimal solution $R^* \in I$ of (8) even in central form is NP-Hard [28]. The computational time of finding the optimizer set increases exponentially with $N$ [29]. The well-known sequential greedy algorithm finds a suboptimal solution $R_{SG}$ for (8) with the optimality bound of $f(R_{SG}) \geq \frac{1}{2} f(R^*)$, i.e., a 1/2-approximation at worst case [28]. More recently, by using the multilinear extension of the submodular utility functions, Vondrak [21] developed a randomized centralized continuous greedy algorithm which achieves a $(1 - 1/e) - O(1/T)$-approximate for set value optimization [3] in the value oracle model, see Algorithm 1. Our objective in this paper is to develop a distributed implementation of Algorithm 1 to solve (8) for when agents interact over a connected graph $G$. Recall that in our problem setting, every agent $i \in A$ can evaluate the utility function for a given $R \subset P$ but it has access only to its own policy space $P_i$.

5 A polynomial-time distributed multi-agent randomized continuous greedy algorithm

Our proposed distributed multi-agent randomized continuous greedy algorithm to solve the set value optimization problem [3] over a connected graph $G$ is given in Algorithm 2, whose convergence

\[\text{Pipage rounding is a polynomial time algorithm which moves a fractional point } \hat{x} \text{ on a hypercube to an integral point } x \text{ on the same hypercube such that } f(\hat{x}) \geq f(x)\]
Let $f$ be a submodular function following [21]. We also provide some intermediate results that will be useful in establishing our results.

### 5.1 A short overview of the central continuous greedy process

As we mentioned earlier the continuous multilinear extension is a relaxation strategy that extends a submodular function $f(\mathcal{R})$, which is defined on the vertices of the $n$-dimensional hypercube $\{0, 1\}^n$ to a continuous multilinear function $F$ defined on $[0, 1]^n$. The two functions evaluate identically for any vector $x \in [0, 1]^n$ that is the membership indicator vector of a set $\mathcal{R} \in \mathcal{P}$. Then, by way of a process that runs continuously, depending only on local properties of $F$, we can produce a point $x \in P(\mathcal{M})$ to approximate the optimum $OPT = \max_{x \in P(\mathcal{M})} F(x)$ (here recall (7)). The proposal in [21] is to move in the direction of a vector constrained by $P(\mathcal{M})$ which maximizes the local gain.

To understand the logic behind Algorithm 1 let us review the conceptual steps of this continuous greedy process. [21] views the process as a particle starting at $x(0) = 0$ and following the flow

$$\frac{dx}{dt} = v(x) \quad \text{where} \quad v(x) = \arg \max_{v \in P(\mathcal{M})} v.\nabla F(x).$$

over a unit time interval $[0, 1]$. We note that $x(t)$ for $t \in [0, 1]$ is contained in $P(\mathcal{M})$, since it is a convex combination of vectors in $P(\mathcal{M})$. Now consider a point $x$ along the trajectory of our flow (9) and assume that $x^*$ is the true optimum $OPT = F(x^*)$. Now consider a direction $v^* = \max\{x^* - x, 0\}$, which is a nonnegative vector. Because $0 \leq v^* \leq x^* \in P(\mathcal{M})$, then $v^* \in P(\mathcal{M})$. By virtue of Lemma 3.1, $F$ is monotone increasing, therefore, using $\max\{x^* - x, 0\} = \max\{x^*, x\} - x$ we have $F(x + v^*) = F(\max\{x^*, x\}) \geq F(x^*)$. However, $x + v^*$ does not belong to $P(\mathcal{M})$ necessarily. Therefore, let's consider $F(x + \zeta v^*)$ for $\zeta \geq 0$. It can be shown that $F(x + \zeta v^*)$ is concave in $\zeta$ and $\frac{dF}{d\zeta}|_{\zeta=0} = v^*.\nabla F(x)$. But, since $v^* \in P(\mathcal{M})$, and $v \in P(\mathcal{M})$ that is used to generate $v$ maximizes $v.\nabla F(x)$, we can write $v.\nabla F(x) \geq v^*.\nabla F(x) \geq F(x + v^*) - F(x) \leq OPT - F(x)$. Now we note that $\frac{dF}{d\zeta} = v(x).\nabla F(x) \geq OPT - F(x(t))$. Therefore, given $x(0) = 0$, using the Comparison Lemma [30], we can conclude the $F(x) \geq (1 - e^{-t})OPT$, and thus $x(1) \in P(\mathcal{M})$ and also $F(x(1)) \geq (1 - 1/e)OPT$. In the second stage of the algorithm, the fractional solution $x(1)$ is rounded to a point in $\{0, 1\}^n$ by use of Pipage rounding method, see [31] for more details about Pipage rounding. The aforementioned exposition is the conceptual design behind the continuous greedy process. The algorithm 1 is a practical implementation achieved by the use of a numerical iterative process

$$x(t+1) = x(t) + \frac{1}{T}v(t),$$

and use of sampling to compute $\nabla F(x)$ and consequently $v(t)$, see [21] for more details. In what follows, we explain a practical distributed implementation of the continuous greedy process, which is realized as Algorithm 2 and is inspired by this central solution.

### 5.2 Design and analysis of the distributed continuous greedy process

We start off our design and analysis of the distributed continuous greedy process by introducing our notation and the set of computations that agents carry out locally using their local information and interaction with their neighbors. The algorithm is performed over the finite time horizon of $T$ steps. Let $\mathcal{F}_i(t) \subset \mathcal{P} \times [0, 1]$ be the information set of agent $i$ at time step $t$, initialized at $\mathcal{F}_i(0) = \emptyset$. For, any couple $(p, \alpha) \in \mathcal{F}_i(t)$ the first element is the policy and the second element is the corresponding membership probability. We let $x_{i}(t) \in \mathbb{R}^n$ (recall $|\mathcal{P}| = n$) be the local copy of the membership probability of our suboptimal solution of (8) at agent $i$ at time step $t$, defined according to

$$x_{ip}(t) = \begin{cases} \alpha, & (p, \alpha) \in \mathcal{F}_i(t), \\ 0, & \text{otherwise}, \end{cases} \quad p \in \mathcal{P}. \quad (11)$$
Algorithm 2 Discrete Distributed implementation of the continuous greedy process.

1: Init: $\bar{F} \leftarrow \emptyset$, $\mathcal{F}_i \leftarrow \emptyset$, $t \leftarrow 1$.

2: while $t \leq T$ do

3: for $i \in \mathcal{A}$ do

4: Draw $K_i$ sample policy sets $\mathcal{R}$ such that $q \in \mathcal{R}$ with the probability $\alpha$ for all $(q, \alpha) \in \mathcal{F}_i$.

5: for $p \in \mathcal{P}_i$ do

6: Compute $w_p^i \sim \mathbb{E}[f(\mathcal{R} \cup \{p\}) - f(\mathcal{R} \setminus \{p\})]$ using the policy sample sets of step 4.

7: end for

8: Solve for $p^* = \arg\max_{p \in \mathcal{P}_i} w_p^i$.

9: $\mathcal{F}_i \leftarrow \mathcal{F}_i \cup \{p^*, \frac{1}{T}\}$.

10: Broadcast $\mathcal{F}_i$ to the neighbors $\mathcal{N}_i$.

11: $\mathcal{F}_i \leftarrow \max_{j \in \mathcal{N}_i \cup \{i\}} \mathcal{F}_j$.

12: end for

13: $t \leftarrow t + 1$.

14: end while

15: for $i \in \mathcal{A}$ do

16: Sample one policy $\bar{p} \in \mathcal{P}_i$ using $\mathcal{F}_i$

17: $\mathcal{P} \leftarrow \mathcal{P} \cup \{\bar{p}\}$

18: end for

Recall that $\mathcal{P} = \{1, \ldots, n\}$ and it is sorted agent-wise with $1 \in \mathcal{P}_1$ and $n \in \mathcal{P}_N$. Hence, $x_i(t) = [\hat{x}_{i1}(t), \ldots, x_{it}(t), \ldots, \hat{x}_{in}(t)]^T$ where $x_{ij}(t) \in \mathbb{R}_{\geq 0}^{P_i}$ is the membership probability vector of agent $i$’s own policy at iteration $t$, while $\hat{x}_{ij}(t) \in \mathbb{R}_{\geq 0}^{P_i}$ is the local estimate of the membership probability vector of agent $j$ by agent $i$. At time step $t$ agent $i$ solves the optimization problem

$$\tilde{x}_i(t) = \arg\max_{y \in P_i(M)} y\nabla F(x_i(t))$$

with

$$P_i(M) = \left\{ [y_1^T, \ldots, y_N^T]^T \in \mathbb{R}_{\geq 0}^n \mid 1^T y_i \leq 1, y_j = 0, j \in \mathcal{A} \setminus \{i\} \right\}. \quad (13)$$

The term $\nabla F(x_i(t))$ is the estimate of $\nabla F(x_i(t))$ which is calculated by taking $K_i$ samples of set $\mathcal{R}_{x_i(t)}$ according to membership probability vector $x_i(t)$. Recall (5). Hence, $\frac{\partial F}{\partial x_p}$ is estimated by averaging $f(\mathcal{R}_{x_i(t)} \cup \{p\}) - f(\mathcal{R}_{x_i(t)} \setminus \{p\})$ over the samples. We denote the $p$th element of $\nabla F(x_i(t))$ by $w_p^i$, and represent it by

$$w_p^i \sim \mathbb{E}[f(\mathcal{R}_{x_i(t)} \cup \{p\}) - f(\mathcal{R}_{x_i(t)} \setminus \{p\})]. \quad (14)$$

We note that given the definition of $P_i(M)$, to compute (12), we only need $w_p^i$ for $p \in \mathcal{P}_i$.

Remark 5.1 (Local computation of $w_p^i$, $p \in \mathcal{P}_i$, by agent $i$) Given the definition (11), we note that $w_p^i$, an estimate of $\frac{\partial F}{\partial x_p}$, can be obtained from drawing $K_i$ sample policy sets $\mathcal{R}$ such that $q \in \mathcal{R}$ with the probability $\alpha$ for all $(q, \alpha) \in \mathcal{F}_i$ and using

$$w_p^i \sim \mathbb{E}[f(\mathcal{R} \cup \{p\}) - f(\mathcal{R} \setminus \{p\})], \quad p \in \mathcal{P}_i. \quad (15)$$

Let each agent propagate its local variable according to

$$x_i^-(t + 1) = x_i(t) + \frac{1}{T} \tilde{x}_i(t). \quad (16)$$

One can make a conceptual connection between (16) and the practical implementation of (9) discussed earlier. Because the propagation is only based on local information of agent $i$, next, each
agent, by interacting with its neighbors, updates its propagated $x_i^-(t+1)$ by element-wise maximum seeking

$$x_i(t+1) = \max_{j \in \mathcal{N}_i \cup \{i\}} x_j^-(t+1). \quad (17)$$

Lemma 5.1 below shows that, as one expects, $x_{ii}(t+1) = x_{ii}^-(t+1)$, i.e., the corrected component of $x_i$ corresponding to agent $i$ itself is the propagated value maintained at agent $i$, and not the estimated value of any of its neighbors.

**Lemma 5.1** Assume that the agents follow the distributed Algorithm 2. Let $\bar{x}(t) = \max_{i \in \mathcal{A}} x_i(t)$ where $x_i$ is given in (11). Moreover, Then, $\bar{x}(t) = [x_{11}(t), \ldots, x_{NN}(t)]^\top$ at any time step $t$.

The proof of Lemma 5.1 is given in Appendix A. We note that it follows from Lemma 5.1 that

$$x_{jj}(t) = x_{jj}^-(t), \quad j \in \mathcal{A}. \quad (18)$$

In the distributed Algorithm 2 at each time step $t$ each agent has its own belief on the probabilities of the policies that is not necessarily the same as the belief of the other agents. The following result establishes the difference between the belief of the agents.

**Proposition 5.1** Let agents follow the distributed Algorithm 2. Then, the vectorized membership probability $x_i(t)$ for each agent $i \in \mathcal{A}$ realized from $\mathcal{F}_i(t)$ satisfy

$$0 \leq \frac{1}{N} 1_{\bar{\mathcal{F}}}(\bar{x}(t) - x_i(t)) \leq \frac{1}{T} d(\mathcal{G}), \quad (19a)$$

$$\bar{x}(t+1) - \bar{x}(t) = \frac{1}{T} \sum_{i \in \mathcal{A}} \bar{v}_i(t), \quad (19b)$$

$$\frac{1}{N} 1_{\bar{\mathcal{F}}}(\bar{x}(t+1) - \bar{x}(t)) = \frac{1}{T}. \quad (19c)$$

The proof of Proposition 5.1 is given in the Appendix A. Because $f$ is normal and monotone increasing, we have the guarantees that $w_p^i \geq 0$. Therefore, without loss of generality, we know that one realization of $\bar{v}_i(t)$ in (12) corresponds to $1_{1(p^*)}$ where

$$p^* = \arg \max_{p \in \mathcal{P}_i} w_p^i. \quad (20)$$

Next, let each agent $i \in \mathcal{A}$ propagate its information set according to

$$\mathcal{F}_i^-(t+1) = \mathcal{F}_i(t) \oplus \left\{ p^*, \frac{1}{T} \right\}, \quad (21)$$

and update it using a local interaction with its neighbors according to

$$\mathcal{F}_i(t+1) = \max_{j \in \mathcal{N}_i \cup \{i\}} \mathcal{F}_j^- (t+1). \quad (22)$$

By definition to $\oplus$ and MAX operators, we have the guarantees that if $(p, \alpha_1)$, then there exists no $\alpha_2 \neq \alpha_1$ that $(p, \alpha_2) \in \mathcal{F}_i$.

**Lemma 5.2** For $\bar{v}_i(t) = 1_{1(p^*)}$. $x_i^-(t+1)$ and $x_i(t+1)$ computed from, respectively, (16) and (17) are the same as $x_i^-(t+1)$ and $x_i(t+1)$ constructed from, respectively, $\mathcal{F}_i^-(t+1)$ and $\mathcal{F}_i(t+1)$ using (11).

**Proof of Lemma 5.2** The proof follows trivially from the definition of the operator $\oplus$ and (11). □

Initialized by $\mathcal{F}_i(0) = \emptyset$, $i \in \mathcal{A}$, (20), (21), and (22) where $w_p^i$ is computed via (15) constitute a distributed iterative process, formally stated by Algorithm 2 that runs for $T$ steps. At the end
of these $T$ steps, as stated in Algorithm 2, each agent $i \in A$, obtains its suboptimal policy $\bar{P}_i$ by sampling one policy $\bar{p} \in P_i$ with the probability given by $x_{ip}(T)$, where for $p \in P_i$, 

$$x_{ip}(T) = \begin{cases} \alpha, & (p, \alpha) \in F_i(T), \\ 0 & \text{otherwise.} \end{cases}$$

The following result, whose proof is available in the Appendix A, gives the convergence guarantee and suboptimality gap of Algorithm 2.

**Theorem 5.1 (Convergence guarantee and suboptimality gap of Algorithm 2)** Let $f : 2^P \to \mathbb{R}_{\geq 0}$ be normalized, monotone increasing and submodular set function. Let $S^*$ be the optimizer of problem (8). Then, the admissible policy set $\bar{P}$, the output of distributed Algorithm 2 satisfies

$$\left(1 - \frac{1}{e}\right) \left(1 - \left(2N^2d(G) + \frac{1}{2} N^2 + N\right) \frac{1}{T}\right) f(S^*) \leq \mathbb{E}[f(\bar{P})],$$

with the probability of $\left(\prod_{i \in A} (1 - 2e^{-\frac{1}{T^2} K_i})^{\vert P_i \vert}\right)^T$.

By simplifying the probability statement and dropping the higher order terms, the optimality gap grantee of Theorem 5.1 holds with the probability of at least

$$\left(1 - \frac{1}{e}\right) \left(1 - \left(2N^2d(G) + \frac{1}{2} N^2 + N\right) \frac{1}{T}\right) f(S^*) \leq \mathbb{E}[f(\bar{P})].$$

Remark 5.2 (Extra communication for improved optimality gap) Replacing the update step (17) with $x_i(t+1) = y_i(d(G))$ where $y_i(0) = x_i^- (t + 1)$ and

$$y_i(m) = \max_{j \in N_i \cup \{i\}} y_j(m-1), \quad m \in \{1, \cdots, d(G)\},$$

i.e., starting with $x_i^- (t + 1)$ and recursively repeating the update step (17) using the output of the previous recursion for $d(G)$ times, each agent $i \in A$ arrives at $x_i(t+1) = \bar{x}(t+1)$ (recall Lemma 5.1). Hence, for this revised implementation, following the proof of Theorem 5.1 we observe that (33) is replaced by

$$\left(1 - \frac{1}{e}\right) \left(1 - \left(2N^2d(G) + \frac{1}{2} N^2 + N\right) \frac{1}{T}\right) f(S^*) \leq \mathbb{E}[f(\bar{P})],$$

with the probability of $\left(\prod_{i \in A} (1 - 2e^{-\frac{1}{T^2} K_i})^{\vert P_i \vert}\right)^T$. This improved optimality gap is achieved by $(d(G) - 1)T$ extra communication per agent. The optimality bound (23) is the same bound that is achieved by the centralized algorithm of (27). To implement this revision, Algorithm 2’s step 11 (equivalent to (21)) should be replaced by $\bar{F}_i = H_i(d(G))$, where $H_i(0) = F_i^-$, and

$$H_i(m) = \max_{j \in N_i \cup \{i\}} H_j^- (m-1), \quad m \in \{1, \cdots, d(G)\}. \quad (24)$$

6 Numerical Example

Consider the multi-agent sensor placement problem introduced in Fig. 1 for 5 agents for which $B_i = B$, i.e., each agent is able to move to any of the sensor placement points. This sensor placement problem is cast by optimization problem (8). The field is a 6 unit by 6 unit square and the feasible sensor locations are the 6 by 6 grid in the center square of the field, see Fig. 2. The points of interest are spread around the map (small red dots in Fig. 2) in the total number of 900. The sensing zone of the agents $A = \{a, b, c, d, e\}$ are circles with radii of respectively $\{0.5, 0.6, 0.7, 0.8, 1.5\}$. The agents communicate over a ring graph as shown in Fig. 2. We first solve this problem using our
Figure 2: Plots (a)-(f) show 6 different SEQs used in the sequential greedy algorithm. Plot (g) shows the outcome of using Algorithm 2 whereas plot (h) shows the outcome of the sequential greedy algorithm when SEQ in Case 1 (plot (a)) is used.

Table 1: The outcome of Algorithm 2 for different iteration and sampling numbers.

| $T$  | 10000 | 500  | 100  | 50  | 10  | 5  | 1  |
|------|-------|------|------|-----|-----|----|----|
| 100  | 768   | 768  | 718  | 710 | 716 | 716| 696|
| 20   | 768   | 768  | 718  | 710 | 726 | 716| 696|
| 10   | 661   | 640  | 657  | 640 | 634 | 602| 551|
| 5    | 630   | 630  | 634  | 626 | 583 | 608| 540|
| 1    | 456   | 456  | 456  | 456 | 456 | 456| 456|

The results of the simulation for different iteration and sampling numbers are shown in Table 1. The algorithm produces good results at a modest number of iteration and sampling numbers (e.g. see $T = 20$ and $K = 500$). Fig. 2(g) shows the result of the deployment using Algorithm 2. Next, we solve this algorithm using the sequential greedy algorithm [29] in a decentralized way by first choosing a route $SEQ = a \rightarrow b \rightarrow c \rightarrow d \rightarrow e$ that visits all the agents, and then giving $SEQ$ to the agents so they follow $SEQ$ to share their information in a sequential manner. Figure 2(a)-(f) gives 6 possible $SEQ$ denoted by the semi-circular arrow inside the networks.

What stands out about the sequential greedy algorithm is that the choice of sequence
Table 2: Outcome of sequential greedy algorithm.

| Case | 1   | 2   | 3   | 4   | 5   | 6   |
|------|-----|-----|-----|-----|-----|-----|
| Utility | 634 | 704 | 699 | 640 | 767 | 760 |

Figure 3: The sequential greedy algorithm, when the blue agent chooses first assigns both the blue and the orange agents to point A resulting in inferior performance compared to the case that the orange agent chooses first. In the later case, orange agent gets A and the blue agent gets B, which is indeed the optimal solution.

can affect the outcome of the algorithm significantly. We can attribute this inconsistency to the heterogeneity of the sensors’ measurement zone. We can see that when sensor $e$ is given the last priority to make its choice, the sequential greedy algorithm acts poorly. This can be explained by agents with smaller sensing zone picking high-density areas but not being able to cover it fully, see Fig. 3(h) which depicts the outcome of a sequential greedy algorithm using the sequence in Case 1. A simpler example justifying this observation is shown in Fig. 3 with the two disjoint clusters of points and two sensors. One may suggest to sequence the agents from high to low sensing zone order, however this is not necessarily the best choice as we can see in Table 2 the utility of case 6 is less than case 5 (the conjecture of sequencing the agents from strongest to weakest is not valid). Moreover, this ordering may lead to a very long $SEQ$ over the communication graph. Interestingly, this inconsistency does not appear in solutions of Algorithm 2 where the agents intrinsically are overcoming the importance of a sequence by deciding the position of the sensors over a time horizon of communication and exchanging their information set.

7 Conclusion

We proposed a distributed suboptimal algorithm to solve the problem of maximizing an monotone increasing submodular set function subject to a partitioned matroid constraint. Our problem of interest was motivated by optimal multi-agent sensor placement problems in discrete space. Our algorithm was a practical decentralization of a multilinear extension based algorithm that achieves $(1 - 1/e - O(1/T))$ optimally gap, which is an improvement over $1/2$ optimality gap that the well-known sequential greedy algorithm achieves. In our numerical example, we compared the outcome obtained by our proposed algorithm with that of a decentralized sequential greedy algorithm which is constructed from assigning a priority sequence to the agents. We showed that the outcome of the sequential greedy algorithm is inconsistent and depends on the sequence. However, our algorithm’s outcome due to its iterative nature intrinsically tended to be consistent, which in some ways also explains its better optimally gap over the sequential greedy algorithm. Our future work is to study the robustness of our proposed algorithm to message dropout.

Acknowledgments

This work is supported by NSF award IIS-SAS-1724331.
References

[1] S. Martínez and F. Bullo, “Optimal sensor placement and motion coordination for target tracking,” *Automatica*, vol. 42, no. 4, pp. 661–668, 2006.

[2] N. Zhou, C. Cassandras, X. Yu, and S. Andersson, “Decentralized event-driven algorithms for multi-agent persistent monitoring tasks,” in *IEEE Int. Conf. on Decision and Control*, pp. 4064–4069, 2017.

[3] Y. Khazaeni and C. Cassandras, “Event-driven trajectory optimization for data harvesting in multi-agent systems,” *IEEE Tran. on Control of Network Systems*, 2017.

[4] N. Zhou, X. Yu, S. Andersson, and C. Cassandras, “Optimal event-driven multi-agent persistent monitoring of a finite set of data sources,” *IEEE Tran. on Automatic Control*, 2018.

[5] N. Rezazadeh and S. S. Kia, “A sub-modular receding horizon approach to persistent monitoring for a group of mobile agents over an urban area,” *IFAC-PapersOnLine*, vol. 52, no. 20, pp. 217–222, 2019.

[6] X. Duan, M. George, R. Patel, and F. Bullo, “Robotic surveillance based on the meeting time of random walks,” *IEEE Tran. on Robotics and Automation*, 2020.

[7] S. Welikala and C. Cassandras, “Event-driven receding horizon control for distributed estimation in network systems,” *arXiv preprint arXiv:2009.11958*, 2020.

[8] S. Alamdari, E. Fata, and L. Smith, “Persistent monitoring in discrete environments: Minimizing the maximum weighted latency between observations,” *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 138–154, 2014.

[9] N. Mehr and R. Horowitz, “A submodular approach for optimal sensor placement in traffic networks,” in *2018 Annual American Control Conference (ACC)*, pp. 6353–6358, IEEE, 2018.

[10] H. A, M. Ghasemi, H. Vikalo, and U. Topcu, “Randomized greedy sensor selection: Leveraging weak submodularity,” *IEEE Transactions on Automatic Control*, 2020.

[11] V. Tzoumas, A. Jadbabaie, and G. J. Pappas, “Sensor placement for optimal kalman filtering: Fundamental limits, submodularity, and algorithms,” in *American Control Conference*, pp. 191–196, IEEE, 2016.

[12] M. Shamaiah, S. Banerjee, and H. Vikalo, “Greedy sensor selection: Leveraging submodularity,” in *IEEE conference on decision and control*, pp. 2572–2577, 2010.

[13] S. T. Jawaid and S. Smith, “Submodularity and greedy algorithms in sensor scheduling for linear dynamical systems,” *Automatica*, vol. 61, pp. 282–288, 2015.

[14] G. Nemhauser, L. Wolsey, and M. Fisher, “An analysis of approximations for maximizing submodular set functions—i,” *Mathematical programming*, vol. 14, no. 1, pp. 265–294, 1978.

[15] L. Fisher, G. Nemhauser, and L. Wolsey, “An analysis of approximations for maximizing submodular set functions—ii,” in *Polyhedral combinatorics*, pp. 73–87, Springer, 1978.

[16] K. Wei, R. Iyer, and J. Bilmes, “Fast multi-stage submodular maximization,” in *International conference on machine learning*, pp. 1494–1502, PMLR, 2014.

[17] B. Mirzasoleiman, A. Karbasi, R. Sarkar, and A. Krause, “Distributed submodular maximization: Identifying representative elements in massive data,” in *Advances in Neural Information Processing Systems*, pp. 2049–2057, 2013.

[18] B. Mirzasoleiman, M. Zadimoghaddam, and A. Karbasi, “Fast distributed submodular cover: Public-private data summarization,” in *Advances in Neural Information Processing Systems*, pp. 3594–3602, 2016.

[19] R. Kumar, B. Moseley, S. Vassilvitskii, and A. Vattani, “Fast greedy algorithms in mapreduce and streaming,” *ACM Transactions on Parallel Computing*, vol. 2, no. 3, pp. 1–22, 2015.

[20] P. S. Raut, O. Sadeghi, and M. Fazel, “Online dr-submodular maximization with stochastic cumulative constraints,” *arXiv preprint arXiv:2005.14708*, 2020.
Appendix A: Proof of the Results in Section 5

Proof of Lemma 5.1 Since $f$ is a monotone increasing and submodular set function, we have $f(R_{x_i}(t) \cup \{p\}) - f(R_{x_i}(t) \setminus \{p\}) \geq 0$ and hence $\nabla F(x_i(t))$ has positive entries $\forall i \in \mathcal{A}$. This results in the optimization (12) subject to vector space (13) to output vector $\tilde{v}_i(t) \in P_i(M)$ that has entries greater or equal to zero. Hence, according to the propagation and update rule (16) and (17), we can conclude that $x_{ii}(t)$ has increasing elements and only agent $i$ can update it and other agents only copy this value as $\hat{x}_{ji}(t)$. Therefore, we can conclude that $\hat{x}_{ijp}(t) \leq x_{ijp}(t)$ for all $p \in P_i$ which concludes the proof.

Proof of Proposition 5.1 If $f$ is a monotone increasing and submodular set function therefor $f(R_{x_i}(t) \cup \{p\}) - f(R_{x_i}(t) \setminus \{p\}) \geq 0$ and hence $\nabla F(x_i(t))$ has positive entries $\forall i \in \mathcal{A}$. Then, because $\tilde{v}_j(t) \in P_j(M)$, it follows from (12) that $\tilde{v}_j(t)$ has non-negative entries, $\tilde{v}_{jp}(t) \geq 0$, which satisfy $\sum_{p \in P_j} \tilde{v}_{jp}(t) = 1$. Therefore, it follows from (16) and Lemma 5.1 that

$$1.x_{jj}(t+1) = 1.x_{jj}(t) + \frac{1}{T}, \quad j \in \mathcal{A}. \quad (25)$$

Using (25), we can also write

$$1.x_{jj}(t) = 1.x_{jj}(t-d(G)) + \frac{1}{T}d(G), \quad j \in \mathcal{A}. \quad (26)$$
Furthermore, it follows from Lemma 5.1 that for all \( p \in \mathcal{P}_j \) and any \( i \in \mathcal{A}\setminus\{j\} \) we can write
\[
{x_{ijp}(t)} \geq \dot{x}_{ijp}(t).
\] (27)

Also, since every agent \( i \in \mathcal{A}\setminus\{j\} \) can be reached from agent \( j \in \mathcal{A} \) at most in \( d(\mathcal{G}) \) hops, it follows from the propagation and update laws [16] and [17], for all \( p \in \mathcal{P}_j \), for any \( i \in \mathcal{A}\setminus\{j\} \) that
\[
\dot{x}_{ijp}(t) \geq x_{ijp}(t - d(\mathcal{G})).
\] (28)

Thus, for any \( j \in \mathcal{A} \) and \( i \in \mathcal{A}\setminus\{j\} \), (27) and (28) result in
\[
1. x_{jj}(t) \geq 1. \dot{x}_{ij}(t) \geq 1. x_{ij}(t - d(\mathcal{G})).
\] (29)

Next, we can use (26) and (29) to write
\[
1. x_{jj}(t) \geq 1. \dot{x}_{ij}(t) \geq 1. x_{jj}(t) - \frac{1}{T} d(\mathcal{G}),
\] (30)

for \( j \in \mathcal{A} \) and \( i \in \mathcal{A}\setminus\{j\} \). Using (30) for any \( i \in \mathcal{A} \) we can write
\[
\sum_{j \in \mathcal{A}} 1. x_{jj}(t) \geq x_{ii}(t) + \sum_{j \in \mathcal{A}\setminus\{i\}} 1. \dot{x}_{ij}(t) \geq \sum_{j \in \mathcal{A}} 1. x_{jj}(t) - \frac{1}{T} Nd(\mathcal{G}).
\] (31)

Then, using Lemma 5.1 from (31) we can write
\[
\dot{x}(t) \geq 1. x(t) \geq 1. \dot{x}(t) - \frac{1}{T} Nd(\mathcal{G}),
\]
which ascertains [19a]. Next, note that from Lemma 5.1 we have \( x_{jj}(t) = x_{ij}(t) \) for any \( j \in \mathcal{A} \). Then, using [16] and invoking Lemma 5.1 we obtain (19b), which, given (25), also ascertains (19c).

**Proof of Theorem 5.1** Knowing that \( \left| \frac{\partial^2 F}{\partial x_p \partial x_q} \right| \leq f(S^*) \) from Lemma 7.2 and (19c), it follows from Lemma 7.3 that
\[
F(\bar{x}(t + 1)) - F(\dot{x}(t)) \geq \nabla F(\dot{x}(t)).(\ddot{x}(t + 1) - \dot{x}(t)) - \frac{1}{2} N^2 \frac{1}{T^2} 1 f(S^*),
\]
which, given (19b), leads to
\[
F(\bar{x}(t + 1)) - F(\dot{x}(t)) \geq \frac{1}{T} \sum_{i \in \mathcal{A}} 1 \ddot{x}_i(t). \nabla F(\dot{x}(t)) - \frac{1}{2} N^2 \frac{1}{T^2} 1 f(S^*).
\] (32)

Next, we note that by definition, \( \ddot{x}(t) \geq x_i(t) \) for any \( i \in \mathcal{A} \). Therefore, given (19b), by invoking Lemma 7.3 for any \( i \in \mathcal{A} \) we can write
\[
\left| \frac{\partial F}{\partial x_p}(\ddot{x}(t)) - \frac{\partial F}{\partial x_p}(x_i(t)) \right| \leq N \frac{1}{T} d(\mathcal{G}) f(S^*),
\] (33)

for \( p \in \{1, \cdots, n\} \). Recall that at each time step \( t \), the realization of \( \ddot{v}_i(t) \) in (12) that Algorithm 2 uses is
\[
\ddot{v}_i(t) = 1_{p^*}, \quad p^* \in \mathcal{P}_i \text{ is given by (20)}
\] (34)

for every \( i \in \mathcal{A} \). Thus, \( 1. \ddot{v}_i(t) = 1, i \in \mathcal{A} \). Consequently, using (33) we can write
\[
\sum_{i \in \mathcal{A}} \ddot{v}_i(t). \nabla F(\dot{x}(t)) \geq \sum_{i \in \mathcal{A}} \ddot{v}_i(t). \nabla F(x_i(t)) - N^2 \frac{1}{T^2} d(\mathcal{G}) f(S^*).
\] (35)

Next, we let
\[
\ddot{v}_i(t) = \arg\max_{\dot{v} \in F_i(\mathcal{M})} \dot{v}. \nabla F(\dot{x}(t))
\]

and
\[
\dddot{v}_i(t) = \arg\max_{\dot{v} \in F_i(\mathcal{M})} \dot{v}. \nabla F(x_i(t)).
\]
Because $f$ is monotone increasing, by virtue of Lemma 3.1 $\frac{\partial F}{\partial x_p} \geq 0$, and as such $\bar{v}_i(t) = 1_{\bar{p}}$ and $\bar{v}_i(t) = 1_{\bar{p}}$, where $\bar{p} = \arg\max_{p \in P_i} \frac{\partial F}{\partial x_p}$. Therefore, using

$\hat{v}_i(t).\nabla F(x_i(t)) \geq \bar{v}_i(t).\nabla F(x_i(t))$

and

$\hat{v}_i(t).\nabla F(x_i(t)) \geq \bar{v}_i(t).\nabla F(x_i(t))$, $i \in A$, and (33) we can also write

$$\sum_{i \in A} \bar{v}_i(t).\nabla F(x_i(t)) \geq \sum_{i \in A} \bar{v}_i(t).\nabla F(x_i(t)) \geq \sum_{i \in A} \bar{v}_i(t).\nabla F(x_i(t)) - N^2 \frac{1}{T} d(G) f(S^*) \tag{36a}$$

$$\sum_{i \in A} \bar{v}_i(t).\nabla F(x_i(t)) \geq \sum_{i \in A} \bar{v}_i(t).\nabla F(x_i(t)) \geq \sum_{i \in A} \bar{v}_i(t).\nabla F(x_i(t)) - N^2 \frac{1}{T} d(G) f(S^*) \tag{36b}$$

On the other hand, by virtue of Lemma 7.4 $\frac{\partial F}{\partial x_p}(x_j(t)), p \in P_j$ that each agent $j \in A$ uses to solve optimization problem (20) (equivalently (12)) satisfies

$$\left| \frac{\partial F}{\partial x_p}(x_j(t)) - \frac{\partial F}{\partial x_p}(x_j(t)) \right| \leq \frac{1}{2T} f(S^*) \tag{37}$$

with the probability of $1 - 2e^{-\frac{1}{8}\pi \sigma^2 K_j}$. Using (36b) and (37), and also that the samples are drawn independently

$$\sum_{i \in A} \bar{v}_i(t).\nabla F(x_i(t)) \geq \sum_{i \in A} \bar{v}_i(t).\nabla F(x_i(t)) - N^2 \frac{1}{T} f(S^*) \tag{38a}$$

$$\sum_{i \in A} \bar{v}_i(t).\nabla F(x_i(t)) \geq \sum_{i \in A} \bar{v}_i(t).\nabla F(x_i(t)) \geq \sum_{i \in A} \bar{v}_i(t).\nabla F(x_i(t)) - N^2 \frac{1}{T} f(S^*) \tag{38b}$$

with the probability of $\prod_{i \in A} (1 - 2e^{-\frac{1}{8}\pi \sigma^2 K_i}) |P_i|$. From (35), (36a), (38a), and (38b) now we can write

$$\sum_{i \in A} \bar{v}_i(t).\nabla F(x_i(t)) \geq \sum_{i \in A} \bar{v}_i(t).\nabla F(x_i(t)) - (2Nd(G) + 1)N \frac{1}{T} f(S^*) \tag{39}$$

with the probability of $1 - 2 \sum_{i \in A} e^{-\frac{1}{8}\pi \sigma^2 K_i}$.

Next, let $v_i^*$ be the projection of $1_{S^*}$ into $P_i(M)$. Knowing that $P_i(M)$s are disjoint sub-spaces of $P(M)$ covering the whole space then we can write

$$1_{S^*} = \sum_{i \in A} v_i^* \tag{40}$$

Then, using (39), (40), and invoking Lemma 7.1 and the fact that $\bar{v}_i(t).\nabla F(\bar{x}(t)) \geq v_i^*(t).\nabla F(\bar{x}(t))$ we obtain

$$\sum_{i \in A} \bar{v}_i(t).\nabla F(\bar{x}(t)) \geq \sum_{i \in A} v_i^*(t).\nabla F(\bar{x}(t)) \geq (2Nd(G) + 1)N \frac{1}{T} f(S^*) =$$

$$1_{S^*}.\nabla F(\bar{x}(t)) - (2Nd(G) + 1)N \frac{1}{T} f(S^*) \geq f(S^*) - F(\bar{x}(t)) - (2Nd(G) + 1)N \frac{1}{T} f(S^*) \tag{41}$$

with the probability of $\prod_{i \in A} (1 - 2e^{-\frac{1}{8}\pi \sigma^2 K_i}) |P_i|$. Hence, using (32) and (41), we conclude that

$$F(\bar{x}(t + 1)) - F(\bar{x}(t)) \geq \frac{1}{T} f(S^*) - F(\bar{x}(t)) - (2Nd(G) + 1)N \frac{1}{T^2} f(S^*) \tag{42}$$
with the probability of $\prod_{i \in A} (1 - 2e^{-\frac{1}{\sqrt{n}}K_i})^{\vert P_i \vert}$.

Next, let $g(t) = f(S^*) - F(\bar{x}(t))$ and $\beta = (2Nd(G)) + \frac{1}{2}N + 1) \frac{N}{T} f(S^*)$, to rewrite (42) as

$$g(t) = g(t+1) \geq \frac{1}{T} (f(S^*) - F(\bar{x}(t))) - \beta = \frac{1}{T} g(t) - \beta.$$  

(43)

Then from inequality (43) we get

$$g(t+1) \leq (1 - \frac{1}{T}) g(t) + \beta$$  

(44)

with the probability of $\prod_{i \in A} (1 - 2e^{-\frac{1}{\sqrt{n}}K_i})^{\vert P_i \vert}$. Solving for inequality (44) at time $T$ yields

$$g(T) \leq (1 - \frac{1}{T})^T g(0) + \beta \sum_{k=0}^{T-1} \left(1 - \frac{1}{T}\right)^k = (1 - \frac{1}{T})^T g(0) + T\beta (1 - (1 - \frac{1}{T})^T)$$  

(45)

with the probability of $\left(\prod_{i \in A} (1 - 2e^{-\frac{1}{\sqrt{n}}K_i})^{\vert P_i \vert}\right)^T$. Substituting back $g(T) = f(S^*) - F(\bar{x}(T))$ and $g(0) = f(S^*) - F(x(0)) = f(S^*)$, in (45) we then obtain

$$(1 - (1 - \frac{1}{T})^T)(f(S^*) - T\beta) = (1 - (1 - \frac{1}{T})^T) (1 - (2Nd(G)) + \frac{1}{2}N + 1) \frac{N}{T} f(S^*) \leq F(\bar{x}(T)),$$  

(46)

with the probability of $\left(\prod_{i \in A} (1 - 2e^{-\frac{1}{\sqrt{n}}K_i})^{\vert P_i \vert}\right)^T$. By applying $\frac{1}{T} \geq (1 - (1 - \frac{1}{T})^T)$, we get

$$(1 - \frac{1}{e})(1 - (2Nd(G)) + \frac{1}{2}N + 1) \frac{N}{T} f(S^*) \leq F(\bar{x}(T)),$$  

(47)

with the probability of $\left(\prod_{i \in A} (1 - 2e^{-\frac{1}{\sqrt{n}}K_i})^{\vert P_i \vert}\right)^T$.

Given (44), from the propagation and update rules (16) and (17) and Lemma 5.1, we can conclude that $1.x_{ii}(T) = 1$. Furthermore by defining $R_{x_{ii}(T)}$ to be a random set where each member if sampled according to $x_{ii}(T)$ and from $P_i$. Since $1.x_{ii}(T) = 1$, we can also define $T_{x_{ii}(T)}$ to be a random set where only one policy is sampled from $P_i$ according to $x_{ii}(T)$, then using Lemma 7.5, we can write

$$F(\bar{x}(T)) = E[f(R_{x(T)})] = E[f(R_{x_{ii}(T)} \cup \cdots \cup R_{x_{NN}(T)})]$$

$$\leq E[f(T_{x_{ii}(T)} \cup R_{x_{ii23}(T)} \cup \cdots \cup R_{x_{NN}(T)})]$$

$$\leq \cdots$$

$$\leq E[f(T_{x_{ii}(T)} \cup \cdots \cup T_{x_{NN}(T)})]$$

which concludes the proof.

\[\square\]

**Appendix B: Auxiliary Results**

In what follows we let $\mathcal{S}^*$ to represent the optimizer of problem (8).

**Lemma 7.1** Consider the set value optimization problem (8). Suppose $f: \mathcal{P} \rightarrow \mathbb{R}_{\geq 0}$ is an increasing and submodular set function and consider its multilinear extension $F: \mathbb{R}_{\geq 0}^n \rightarrow \mathbb{R}_{\geq 0}$. Then $\forall x \in P(M)$

$$1_{\mathcal{S}^*}.\nabla F(x) \geq f(\mathcal{S}^*) - F(x).$$

**Proof** see (21) for the proof.

\[\square\]
**Lemma 7.2** Consider the set value optimization problem \([S] \). Suppose \( f : \mathcal{P} \to \mathbb{R}_{\geq 0} \) is an increasing and submodular set function and consider its multilinear extension \( F : \mathbb{R}_{\geq 0}^n \to \mathbb{R}_{\geq 0} \). Then

\[
\frac{\partial^2 F}{\partial x_p \partial x_q} \leq f(S^*) \quad p, q \in \mathcal{P}.
\]

**Proof** Since \( p \not\in \mathcal{R}_x \cup \{q\} \setminus \{p\} \), therefore by submodularity of \( f \) we can write

\[
0 \leq \Delta f(p|\mathcal{R}_x \cup \{q\} \setminus \{p\}) \leq f(\{q\}),
\]

and

\[
0 \leq \Delta f(p|\mathcal{R}_x \setminus \{p,q\}) \leq f(\{p\}).
\]

Knowing that

\[
\Delta f(p|\mathcal{R}_x \cup \{q\} \setminus \{p\}) = f(\mathcal{R}_x \cup \{p,q\}) - f(\mathcal{R}_x \cup \{q\} \setminus \{p\}),
\]

and

\[
\Delta f(p|\mathcal{R}_x \setminus \{p,q\}) = f(\mathcal{R}_x \cup \{p\}) - f(\mathcal{R}_x \setminus \{p,q\}),
\]

then definition (6) can be written as

\[
\frac{\partial F}{\partial x_p} = \mathbb{E}[\Delta f(p|\mathcal{R}_x \cup \{q\} \setminus \{p\}) - \Delta f(p|\mathcal{R}_x \setminus \{p,q\})].
\]

Putting (48) and (49) together results in

\[
\left| \frac{\partial^2 F}{\partial x_p \partial x_q} \right| \leq f(\{p\}).
\]

where knowing that \( f(\{p\}) \leq f(S^*) \) concludes the proof. \( \square \)

**Lemma 7.3** Consider a twice differentiable function \( F(x) : \mathbb{R}^N \to \mathbb{R} \) which satisfies \( \left| \frac{\partial^2 F}{\partial x_p \partial x_q} \right| \leq \alpha \) for any \( p, q \in \{1, \cdots, N\} \). Then for any \( x_1, x_2 \in \mathbb{R}^N \) satisfying \( x_2 \geq x_1 \) and \( 1.(x_2 - x_1) \leq \beta \) the following holds,

\[
\left| \frac{\partial F}{\partial x_p}(x_1 + \epsilon(x_2 - x_1)) - \frac{\partial F}{\partial x_p}(x_1) \right| \leq \epsilon \alpha \beta, \quad (50a)
\]

\[
F(x_2) - F(x_1) \geq \nabla F(x_1).(x_2 - x_1) - \frac{1}{2} \alpha \beta^2, \quad (50b)
\]

for \( \epsilon \in [0, 1] \).

**Proof** Let \( h_p = [\frac{\partial^2 F}{\partial x_p \partial x_1}, \cdots, \frac{\partial^2 F}{\partial x_p \partial x_N}]^T \). Then, we can write

\[
\left| \frac{\partial F}{\partial x_p}(x_1 + \epsilon(x_2 - x_1)) - \frac{\partial F}{\partial x_p}(x_1) \right| = \left| \int_0^\epsilon h_p(x_1 + \tau(x_2 - x_1)).(x_2 - x_1)d\tau \right|
\]

\[
\leq \int_0^\epsilon \alpha 1.(x_2 - x_1)d\tau = \epsilon \alpha \beta, \quad (51)
\]

Furthermore,

\[
F(x_2) - F(x_1) = \int_0^1 \nabla F(x_1 + \epsilon(x_2 - x_1)).(\tilde{x}(t + 1) - \tilde{x}(t))d\epsilon
\]

\[
\geq \int_0^1 (\nabla F(x_1) - \epsilon \alpha \beta).(x_2 - x_1)d\epsilon = \nabla F(x_1).(x_2 - x_1) - \frac{1}{2} \alpha \beta^2,
\]

with the third line follow from equation (51), which concludes the proof. \( \square \)
Lemma 7.4 Consider the set value optimization problem (5). Suppose $f: \mathcal{P} \to \mathbb{R}_{\geq 0}$ is an increasing and submodular set function and consider its multilinear extension $F: \mathbb{R}^n_{\geq 0} \to \mathbb{R}_{\geq 0}$. Let $\tilde{\nabla}F(x)$ be the estimate of $\nabla F(x)$ that is calculated by taking $K \in \mathbb{R}_{>0}$ samples of set $\mathcal{R}_x$ according to membership probability vector $x$. Then

$$
\left| \frac{\partial F}{\partial x_p}(x) - \frac{\partial F}{\partial x_p}(\mathbf{x}) \right| \geq \frac{1}{2T} f(S^*) \quad p \in \{1, \cdots, n\},
$$

(52)

with the probability of $2e^{-\frac{1}{2T}K}$, for any $T \in \mathbb{R}_{>0}$.

Proof Define the random variable

$$
X = \left( f(\mathcal{R}_x \cup \{p\}) - f(\mathcal{R}_x \setminus \{p\}) \right) - \frac{\partial F}{\partial x_p}(\mathbf{x})/f(S^*),
$$

and assume that agent $j \in \mathcal{A}$ takes $K_j$ samples from $\mathcal{R}_x$ to construct $\{X_k\}_{k=1}^K$ realization of $X$. Since $f$ is a submodular function, then we have $(f(\mathcal{R}_x \cup \{p\}) - f(\mathcal{R}_x \setminus \{p\})) \leq f(S^*)$. Consequently using equation (5), we conclude that $0 \leq X \leq 1$. Hence, using Theorem 3.1 we have

$$
\sum_{k=1}^K X_k \geq \frac{1}{2T} K.
$$

with the probability of $2e^{-\frac{1}{2T}K}$. Hence, the estimation accuracy of $\nabla F(x)$, is given by

$$
\left| \frac{\partial F}{\partial x_p}(x) - \frac{\partial F}{\partial x_p}(\mathbf{x}) \right| \geq \frac{1}{2T} f(S^*) \quad p \in \{1, \cdots, n\},
$$

with the probability of $2e^{-\frac{1}{2T}K}$. \hfill \Box

Lemma 7.5 Suppose $f: \mathcal{P} \to \mathbb{R}_{\geq 0}$ is an increasing and submodular set function and consider $x$ to be a membership probability vector of set $\mathcal{Q} \subset \mathcal{P}$ with $1 \cdot x = 1$. We define $\mathcal{R}_x$ to be the set resulted by sampling independently each member of $\mathcal{Q}$ according to probability vector $x$ and $T_x = \{t\}, \ t \in \mathcal{Q}$ to be a single member set which is chosen according to $x$. The following holds for any random set $S \in \mathcal{P} \setminus \mathcal{Q}$.

$$
\mathbb{E}[f(\mathcal{R}_{x_{t}}, \cup S)] \leq \mathbb{E}[f(\mathcal{T}_{x_{t}} \cup S)].
$$

Proof Defining $\mathcal{R}_x = \{r_1, \cdots, r_o\}$, then we can write

\[
\mathbb{E}[f(\mathcal{R}_x \cup S)] = \mathbb{E}[f(S)] + \sum_{l=1}^{o} \Delta_f(r_l|S \cup \{r_1, \cdots, r_{l-1}\})]
\]

\[
\leq \mathbb{E}[f(S)] + \sum_{r_l \in \mathcal{R}_x} \Delta_f(r_l|S) = \mathbb{E}_S[\mathbb{E}_{\mathcal{R}_x|S}[f(S) + \sum_{r_l \in \mathcal{R}_x} \Delta_f(r_l|S)]]
\]

\[
= \mathbb{E}_S[f(S) + \mathbb{E}_{\mathcal{R}_x|S}[\sum_{r_l \in \mathcal{R}_x} \Delta_f(r_l|S)]] = \mathbb{E}_S[f(S) + \sum_{r_l \in \mathcal{Q}} x_l \Delta_f(r_l|S)]
\]

\[
= \mathbb{E}_S[f(S) + \mathbb{E}_{T_x|S}[^{\Delta_f(t|S)}]] = \mathbb{E}_S[f(T_x \cup S)]
\]

which concludes the proof. \hfill \Box