2021 60th IEEE Conference on Decision and Control (CDC)
December 13-15, 2021. Austin, Texas

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

Navid Rezazadeh and Solmaz S. Kia, *IEEE member, Senior*

2021 60th IEEE Conference on Decision and Control (CDC) | 978-1-6654-3659-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/CDC45484.2021.9682818

Abstract— **This paper studies distributed submodular optimization subject to partition matroid. We work in the value oracle model where the only access of the agents to the utility function is through a black box that returns the utility function value. The agents are communicating over a connected undirected graph and have access only to their own strategy set. As known in the literature, submodular maximization subject to matroid constraints is NP-hard. Hence, our objective is to propose a polynomial-time distributed algorithm to obtain a suboptimal solution with guarantees on the optimality bound. Our proposed algorithm is based on a distributed stochastic gradient ascent scheme built on the multilinear-extension of the submodular set function. We use a maximum consensus protocol to minimize the inconsistency of the shared information over the network caused by delay in the flow of information while solving for the fractional solution of the multilinear extension model. Furthermore, we propose a distributed framework of finding a set solution using the fractional solution. We show that our distributed algorithm results in a strategy set that when the team objective function is evaluated at worst case the objective function value is in $1-1/\mathrm{e}-O(1/T)$ of the optimal solution in the value oracle model where T is the number of communication instances of the agents. An example demonstrates our results.**

I. Introduction

We consider a group of $\mathcal{A}=\{1,...,N\}$ agents with communication and computation capabilities, interacting over a connected undirected graph $\mathcal{G}(\mathcal{A},\mathcal{E})$. Each agent $i\in\mathcal{A}$ has a distinct discrete policy set $\mathcal{P}_i$ and wants to choose at most one policy from $\mathcal{P}_i$ such that a monotone increasing and submodular utility function $f:2^{\mathcal{P}}\to\mathbb{R}_{\geq0}$, $\mathcal{P}=\bigcup_{i\in\mathcal{A}}\mathcal{P}_i$, evaluated at all the agents' policy selection is maximized[1]. In other words, the agents aim to solve in a distributed manner the discrete domain optimization problem

$$\max_{\mathcal{R}\in\mathcal{I}} f(\mathcal{R}) \tag{1a}$$

$$\mathcal{I}=\big\{\mathcal{R}\subset\mathcal{P}\,\big|\,|\mathcal{R}\cap\mathcal{P}_i|\leq1,\ \forall i\in\mathcal{A}\big\}. \tag{1b}$$

The agents' access to the utility function is through a black box that returns $f(\mathcal{R})$ for any given set $\mathcal{R}\in\mathcal{P}$ (value oracle model). The constraint set (1b) is a partition matroid, which restricts the number of policy choices of each agent $i\in\mathcal{A}$ to one. Many sensor placements for monitoring and coverage can be formulated as problem (1) [1]–[3]; see Fig. 1 for an illustration. Our goal is to design a polynomial-time distributed solution for (1) with formal guarantees on the optimality bound.

The authors are with the Mechanical and Aerospace Eng. Dept. of University of California Irvine, Irvine, USA. This work is supported by NSF award IIS-SAS-1724331.

[1]We use standard notation, but for clarity we provide a brief description of the notation and the definitions in Section II.

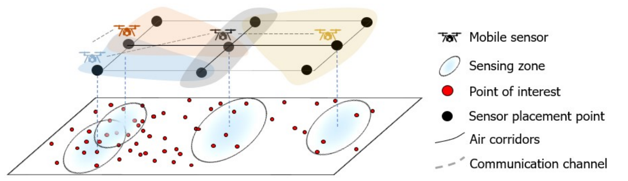


Fig. 1: Let the policy set of each mobile sensor $i\in\mathcal{A}$ be $\mathcal{P}_i=\{(i,p)|p\in\mathcal{B}_i\}$, where $\mathcal{B}_i\subset\mathcal{B}$ is the set of the allowable sensor placement points for agent $i\in\mathcal{A}$ out of all the sensor placement points $\mathcal{B}$. Note that by definition, for any two agent $i,j\in\mathcal{A}$, $\mathcal{P}_i\cap\mathcal{P}_j=\emptyset$. 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 $\mathcal{B}$ 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 monotone and increasing submodular function of the agent's sensing zone [4]. This sensor placement problem can be formalized as the optimization problem (1). The agents communicate over a connected undirected graph, and their objective is to obtain their respective placement point by interacting only with their communicating neighbors.

Discrete set function maximization problems subject to matroid constraints such as (1) are NP hard [5]. When the objective function is monotone increasing and submodular set function, however, it is well-known that the *sequential greedy algorithm* delivers a polynomial-time suboptimal solution with guaranteed optimality bound of $\frac{1}{2}$ times the optimal utility value for problem (1) [6]. For large-scale submodular maximization problems, to manage the sequential greedy algorithm's computational complexity, reducing the problem's size through approximations is proposed in [7]. Using several processing units to accelerate the sequential greedy algorithm with trading some optimality bounds is also proposed in [8]–[11].

More recently, for problems with partition matroid constraint, a suboptimal solution with a better optimality gap is proposed in [12]–[15] using the multilinear-extension to the continuous domain of a submodular set function. The multilinear-extension of a submodular set function allows transforming the discrete problems such as (1) into a continuous optimization problem with linear constraints. Then, as shown in [12] a gradient ascent algorithm along with a continuous to discrete domain mapping procedure is employed to obtain an improved solution with $1-\frac{1}{\mathrm{e}}$ optimality gap bound for a partition matroid constraint, outperforming the sequential greedy algorithm optimality bound of $1/2$. However, this solution requires a central authority. A distributed multilinear-extension-based continuous algorithm that uses an average consensus scheme to solve (1) is proposed in [16]. However, the proposed algorithm assumes that access to the exact multilinear-extension of the utility function is available, whose calculation is exponentially hard. The algorithm also lacks a final distributed rounding technique to map the final continuous solution to an acceptable discrete domain policy set.

In this paper, motivated by the improved optimality gap of the multilinear-extension-based algorithm, we develop a distributed implementation of the algorithm of [12] over a connected undirected graph to obtain a suboptimal solution for (1). We use the stochastic evaluation of the multilinear-extension of a submodular function to manage the computational cost of constructing the utility function's multilinear-extension and propose a gradient-based algorithm, which uses a maximum consensus message passing scheme over the communication graph. Our algorithm uses a distributed stochastic rounding algorithm that allows each agent to obtain its final suboptimal policy selections. Through rigorous analysis, we show that our proposed distributed algorithm achieves, with a known probability, a $1-\frac{1}{\mathrm{e}}-O(1/T)$ optimality bound, where T is the number of times agents communicated over the network. A numerical example demonstrates our results.

II. Notation and definitions

For a vector $\mathbf{x}\in\mathbb{R}^n$, the pth element of the vector is returned as $[\mathbf{x}]_p$. For a set $\mathcal{P}=\{1,\cdots,n\}$ and a vector $\mathbf{x}\in\mathbb{R}^n_{\geq 0}$ with $0\leq[\mathbf{x}]_p\leq 1$, is referred to as *membership probability vector*, and the set $\mathcal{R}_\mathbf{x}$ is a random set where $p\in\mathcal{P}$ is in $\mathcal{R}_\mathbf{x}$ with the probability $[\mathbf{x}]_p$. Furthermore, for $\mathcal{R}\subset\mathcal{P}$, $\mathbf{1}_\mathcal{R}\in\{0,1\}^n$, referred to as *membership indicator vector*, is the vector whose p^{th} element is 1 if $p\in\mathcal{R}$ and 0 otherwise. Lastly, $\Delta_f(p|\mathcal{R})=f(\mathcal{R}\cup\{p\})-f(\mathcal{R})$ for any $\mathcal{R}\subset\mathcal{P}$ and $p\in\mathcal{P}$. $|x|$ is the absolute value of $x\in\mathbb{R}$. By overloading the notation, we also use $|\mathcal{R}|$ as the cordiality of set $\mathcal{R}$. Given a set $\mathcal{F}\subset\mathcal{X}\times\mathbb{R}$ and an element $(p,\alpha)\in\mathcal{X}\times\mathbb{R}$ we define the addition operator $\oplus$ as $\mathcal{F}\oplus\{(p,\alpha)\}=\{(u,\gamma)\in\mathcal{X}\times\mathbb{R}\,|\,(u,\gamma)\in\mathcal{F},u\neq p\}\cup\{(u,\gamma+\alpha)\in\mathcal{X}\times\mathbb{R}\,|\,(u,\gamma)\in\mathcal{F},u=p\}\cup\{(p,\alpha)\in\mathcal{X}\times\mathbb{R}\,|\,(p,\gamma)\notin\mathcal{F},\gamma\in\mathbb{R}\}$. Given a collection of sets $\mathcal{F}_i\in\mathcal{X}\times\mathbb{R}$, $i\in\mathcal{A}$, we define the max-operation over these collection as $\underset{i\in\mathcal{A}}{\text{MAX}}\,\mathcal{F}_i=\{(u,\gamma)\in\mathcal{X}\times\mathbb{R}|(u,\gamma)\in\bar{\mathcal{F}}\text{ s.t. }\gamma=\max\limits_{(u,\alpha)\in\bar{\mathcal{F}}}\alpha\}$, where $\bar{\mathcal{F}}=\bigcup_{i\in\mathcal{A}}\mathcal{F}_i$.

We denote a graph by $\mathcal{G}(\mathcal{A},\mathcal{E})$ where $\mathcal{A}$ is the node set and $\mathcal{E}\subset\mathcal{A}\times\mathcal{A}$ is the edge set. $\mathcal{G}$ is undirected if and only $(i,j)\in\mathcal{E}$ 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 $\mathcal{N}_i=\{j\in\mathcal{A}|(i,j)\in\mathcal{E}\}$. We also use $d(\mathcal{G})$ to show the diameter of the graph.

A set function $f:2^\mathcal{P}\to\mathbb{R}_{\geq 0}$ is normalized if $f(\emptyset)=0$ and monotone increasing if $f(\mathcal{P}_1)\leq f(\mathcal{P}_2)$ for any $\mathcal{P}_1\subset\mathcal{P}_2\subset\mathcal{P}$. Furthermore, the set function f is submoular if $\Delta_f(p|\mathcal{P}_1)\geq\Delta_f(p|\mathcal{P}_2)$, for any $p\in\mathcal{P}\setminus\mathcal{P}_2$, which shows the diminishing return of a set function.

Throughout this paper, without loss of generality, we assume that the ground set $\mathcal{P}$ is equal to $\{1,\cdots,n\}$. For a submodular function $f:2^\mathcal{P}\to\mathbb{R}_{\geq 0}$, its *multilinear-extension* $F:[0,1]^n\to\mathbb{R}_{\geq 0}$ in the continuous space is

$$F(\mathbf{x})=\sum_{\mathcal{R}\subset\mathcal{P}}f(\mathcal{R})\prod_{p\in\mathcal{R}}[\mathbf{x}]_p\prod_{p\notin\mathcal{R}}(1-[\mathbf{x}]_p),\quad \mathbf{x}\in[0,1]^n, \tag{2}$$

which is a unique multilinear function agreeing with f on the vertices of the hypercube $[0,1]^n$.

Lemma 2.1 (See [12]) *If f is non-decreasing, then $\frac{\partial F}{\partial[\mathbf{x}]_p}\geq 0$ for all $p\in\mathcal{P}$, everywhere in $[0,1]^n$ (monotonicity of F). If f is submodular, then $\frac{\partial^2 F}{\partial[\mathbf{x}]_p[\mathbf{x}]_q}\leq 0$ for all $p,q\in\mathcal{P}$, everywhere in $[0,1]^n$.*

Evidently, $F(\mathbf{x})$ equivalently reads as

$$F(\mathbf{x})=\mathbb{E}[f(\mathcal{R}_\mathbf{x})], \tag{3}$$

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

$$\frac{\partial F}{\partial[\mathbf{x}]_p}(\mathbf{x})=\mathbb{E}\left[f(\mathcal{R}_\mathbf{x}\cup\{p\})-f(\mathcal{R}_\mathbf{x}\setminus\{p\})\right], \tag{4a}$$

$$\begin{aligned}\frac{\partial^2 F}{\partial[\mathbf{x}]_p\partial[\mathbf{x}]_q}(\mathbf{x})=\mathbb{E}\,[&f(\mathcal{R}_\mathbf{x}\cup\{p,q\})-f(\mathcal{R}_\mathbf{x}\cup\{q\}\setminus\{p\})\\&-f(\mathcal{R}_\mathbf{x}\cup\{p\}\setminus\{q\})+f(\mathcal{R}_\mathbf{x}\setminus\{p,q\})],\end{aligned} \tag{4b}$$

$p,q\in\mathcal{P}$.

III. Multilinear-extension-based distributed solution

Our proposed distributed algorithm to solve the set value optimization problem (1) over a connected graph $\mathcal{G}$ is a stochastic multilinear-extension-based iterative greedy solution given by Algorithm 1. This algorithm is constructed based on a suboptimal solution to the continuous extension representation of problem (1) that is formed around the multilinear-extension of the utility function. The design procedure, the convergence guarantee, and the suboptimality gap of Algorithm 1 are presented in the following sections. In the first following section, to provide an insight on the design procedure, we review the idea of the central suboptimal solution proposed in [12].

Without loss of generality, in what follows, we let $\mathcal{P}=\bigcup_{i\in\mathcal{A}}\mathcal{P}_i=\{1,\cdots,n\}$, and we let the local policy set of the agents be ordered according to their label $\{1,\cdots,N\}$, such that $1\in\mathcal{P}_1$, and $n\in\mathcal{P}_N$.

A. Review of the centralized multilinear-extension-based solution

We use the multilinear-extension of submodular set functions to extends the submodular utility function $f:2^\mathcal{P}\to\mathbb{R}_{\geq 0}$, which is defined on the vertices of the $n-$dimensional hypercube $\{0,1\}^n$, to the continuous multilinear function F, given in (2), which is defined on $[0,1]^n$. Next, let $\mathbf{x} = [\mathbf{x}_1^\top, ..., \mathbf{x}_N^\top]^\top$, where $\mathbf{x}_i \in \mathbb{R}_{\geq 0}^{|\mathcal{P}_i|}$ is the membership probability vector of $\mathcal{P}_i$ that defines the probability of choosing policies from $\mathcal{P}_i$ for each agent $i \in \mathcal{A}$. Thus, the matroid polytope

$$\mathcal{M} = \{\mathbf{x} \in \mathbb{R}_{\geq 0}^n \,\big|\, \textstyle\sum_{p \in \mathcal{P}_i} [\mathbf{x}]_p \leq 1, \quad \forall i \in \mathcal{A}\} \tag{5}$$

is the convex hull of the partition matroid constraint (1b) in the space of the membership probability vector. Therefore, $\max_{\mathbf{x}\in\mathcal{M}} F(\mathbf{x})$ can be viewed as the continuous approximation of problem (1). Then, by way of a process that runs continuously, depending only on local properties of F, we can produce a point in $\mathcal{M}$ to approximate the optimum $OPT = \max_{\mathbf{x}\in\mathcal{M}} F(\mathbf{x})$. The proposal is to move in the maximum ascent direction in $\mathcal{M}$ by following the flow

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{v}(\mathbf{x}) \quad \text{where} \quad \mathbf{v}(\mathbf{x}) = \underset{\mathbf{w}\in\mathcal{M}}{\arg\max}(\mathbf{w}.\nabla F(\mathbf{x})). \tag{6}$$

over the time interval $[0,1]$. Note that $\mathbf{x}(t)$ for $t \in [0,1]$ is contained in $\mathcal{M}$, since it is a convex combination of vectors in $\mathcal{M}$. [12] shows that by following (6), we obtain $F(\mathbf{x}(1)) \geq (1 - 1/\mathrm{e})\, OPT$. Because for any $\mathcal{R} \in \mathcal{I}$, $\mathbf{1}_\mathcal{R} \in \mathcal{M}$, the global optimizer of (1), $\mathcal{R}^\star$, satisfies $OPT \geq f(\mathcal{R}^\star)$, then $F(\mathbf{x}(1)) \geq (1 - 1/\mathrm{e})\, f(\mathcal{R}^\star)$. Next, by use of the Pipage rounding method [17], the fractional solution $\mathbf{x}(1)$ is rounded to an integral point $\bar{\mathbf{x}} \in \mathcal{M}$, such that $\bar{\mathbf{x}} = [\bar{\mathbf{x}}_1^\top, ..., \bar{\mathbf{x}}_N^\top]^\top \in \{0,1\}^n$ and $F(\bar{\mathbf{x}}) \geq F(\mathbf{x}(1))$. Note that by definition of $\bar{\mathbf{x}} \in \mathcal{M}$, $\bar{\mathbf{x}}_i \in \{0,1\}^{|\mathcal{P}_i|}$, $i \in \mathcal{A}$, has at most one element with value of one. Hence $\bar{\mathcal{R}} = \mathcal{R}_{\bar{\mathbf{x}}} \in \mathcal{I}$ is a deterministic feasible set that satisfies $f(\bar{\mathcal{R}}) \geq F(\mathbf{x}(1))$, and thus, $f(\bar{\mathcal{R}}) \geq (1 - 1/\mathrm{e}) f(\mathcal{R}^\star)$.

Constructing the gradient $\nabla F(\mathbf{x})$ requires the knowledge of $f(\mathcal{R})$ for all $\mathcal{R} \in 2^\mathcal{P}$, which becomes computationally intractable when the size of ground set $\mathcal{P}$ increases. Computing the continuous flow (6) also adds to the computational cost of the solution. A practical solution is achieved by the numerical iterative process

$$\mathbf{x}(t+1) = \mathbf{x}(t) + \frac{1}{T}\mathbf{v}(t), \tag{7}$$

where $1/T$, $T \in \mathbb{Z}_{>0}$, is the step size used to quantize $[0,1]$ and the use of a stochastic sampling procedure to approximate $\nabla F(\mathbf{x})$. Given the stochastic interpretation (4a), if enough set samples are drawn according to membership probability vector $\mathbf{x}$, we can obtain an estimate of $F(\mathbf{x})$ with a reasonable computational cost. The Chernoff-Hoeffding's inequality [18] can be used to quantify the quality of this estimation given the number of samples. The optimality gap for this practical solution is $1 - 1/\mathrm{e} - O(1/T)$ with the probability of $1 - 2T\, n\, \mathrm{e}^{-\frac{1}{8T^2}K}$ where K is the number of samples.

In what follows, we explain a practical distributed implementation of the continuous greedy process, which is realized as Algorithm 1, and is inspired by this central solution.

B. Design and analysis of the multilinear-extension-based distributed solution

The proposed distributed iterative solution is performed over the finite time horizon of T steps. At time t, each agent $i \in \mathcal{A}$ is assigned a local probability membership vector $\mathbf{x}_i(t) \in \mathbb{R}_{\geq 0}^n$ with $\mathbf{x}_i(0) = \mathbf{0}$. Recall that $\mathcal{P} = \{1, ..., n\}$ and it is sorted agent-wise with $1 \in \mathcal{P}_1$ and $n \in \mathcal{P}_N$. Hence, $\mathbf{x}_i(t) = [\hat{\mathbf{x}}_{i1}^\top(t), \cdots, \mathbf{x}_{ii}^\top(t), \cdots, \hat{\mathbf{x}}_{iN}^\top(t)]^\top$ where $\mathbf{x}_{ii}(t) \in \mathbb{R}_{\geq 0}^{|\mathcal{P}_i|}$ is the membership probability vector of agent $i \in \mathcal{A}$ at iteration t, while $\hat{\mathbf{x}}_{ij}(t) \in \mathbb{R}_{\geq 0}^{|\mathcal{P}_j|}$ is the local estimate of the membership probability vector of agent j by agent i.

At time step t, each agent $i \in \mathcal{A}$ empirically computes gradient vector $\nabla F(\mathbf{x}_i(t)) \in \mathbb{R}_{\geq 0}^n$, defined element-wise as

$$\frac{\partial F}{\partial[\mathbf{x}]_p} = f(\mathcal{R}_{\mathbf{x}_i(t)} \cup \{p\}) - f(\mathcal{R}_{\mathbf{x}_i(t)} \setminus \{p\})$$

by taking K_i samples of $\mathcal{R}_{\mathbf{x}_i(t)}$. We let $\widetilde{\nabla F}(\mathbf{x}_i(t))$ to stand for the empirically evaluated vector $\nabla F(\mathbf{x}_i(t))$. Then, each agent $i \in \mathcal{A}$ *greedily* finds the constrained maximum ascent direction as

$$\tilde{\mathbf{v}}_i(t) = \underset{\mathbf{w}\in\mathcal{M}_i}{\text{argmax}}\; \mathbf{w}.\widetilde{\nabla F}(\mathbf{x}_i(t)), \tag{8a}$$

$$\mathcal{M}_i = \left\{\mathbf{w} \in \mathbb{R}_{\geq 0}^n \,\middle|\, \mathbf{1}^\top.\mathbf{w} \leq 1\ , [\mathbf{w}]_p = 0,\ \forall p \in \mathcal{P}\backslash\mathcal{P}_i\right\}. \tag{8b}$$

Then, agent $i \in \mathcal{A}$ *propagates* its local variable according to

$$\mathbf{x}_i^-(t+1) = \mathbf{x}_i(t) + \frac{1}{T}\tilde{\mathbf{v}}_i(t). \tag{9}$$

Next, agent $i \in \mathcal{A}$, by interacting with its neighbors, updates its propagated $\mathbf{x}_i^-(t+1)$ by element-wise maximum seeking

$$\mathbf{x}_i(t+1) = \max_{j\in\mathcal{N}_i\cup\{i\}} \mathbf{x}_j^-(t+1). \tag{10}$$

The following lemma establishes that $\mathbf{x}_{ii}(t+1) = \mathbf{x}_{ii}^-(t+1)$, $i \in \mathcal{A}$, i.e., the updated component of $\mathbf{x}_i$ corresponding to probability of choosing elements from agent i's own local policy set does not change in the update stage. Thus, after updating the local membership probability vectors according to (10), at each agent $i \in \mathcal{A}$ we have $\hat{\mathbf{x}}_{ij}^\top(t) = \mathbf{x}_{jj}^\top(t)$, $j \in \mathcal{N}_i$.

Lemma 3.1: Assume that the agents propagate and update their local probability membership vectors $\mathbf{x}_i(t)$, $i \in \mathcal{A}$ *according to, respectively,* (9) *and* (10). *Let* $\bar{\mathbf{x}}(t) = \max_{i\in\mathcal{A}} \mathbf{x}_i(t)$. *Then, we have*

$$\bar{\mathbf{x}}(t) = [\mathbf{x}_{11}^\top(t), \cdots, \mathbf{x}_{NN}^\top(t)]^\top, \quad t \in \{0, \cdots, T\}. \tag{11}$$

Moreover, $\bar{\mathbf{x}}(t) \in \mathcal{M}$ at any time step t.

The proof of Lemma 3.1 is given in the extended version of this paper available at [19].

Because of the propagation and update rules (9) and (10), the flow of the updated information in the network between the agents is delayed based on how far they are located in the network with respect to each other. Hence at each time step t, each agent has its own probability membership vector $\mathbf{x}_i(t)$ on the policies that do not necessarily agree with the other agents' probability membership vectors. The following result establishes the bounds on the disagreement of the agents.

Proposition 3.1: Assume that the agents propagate and update their local probability membership vector according to, respectively, (9) *and* (10)*. Then, for the membership probability vector of each agent, $i \in \mathcal{A}$, i.e., $\mathbf{x}_i(t)$, satisfies*

$$0 \leq \frac{1}{N}\mathbf{1}.(\bar{\mathbf{x}}(t)-\mathbf{x}_i(t)) \leq \frac{1}{T}d(\mathcal{G}), \tag{12a}$$

$$\bar{\mathbf{x}}(t+1)-\bar{\mathbf{x}}(t) = \frac{1}{T}\sum_{i\in\mathcal{A}}\widetilde{\mathbf{v}}_i(t) \tag{12b}$$

$$\frac{1}{N}\mathbf{1}.(\bar{\mathbf{x}}(t+1)-\bar{\mathbf{x}}(t)) = \frac{1}{T}, \tag{12c}$$

where $\bar{\mathbf{x}}(t)$ is given in (11).

The proof of Proposition 3.1 is given in the extended version of this paper available at [19]

Let the agents terminate their propagation and update of their probability membership vector according to, respectively, (9) and (10) at T steps. The following lemma quantifies the optimality of $\bar{\mathbf{x}}(T) \in \mathcal{M}$ evaluated by the multilinear-extension function F.

Lemma 3.2 (Optimality gap of decentralized multilinear-extension-based framework) *Let $\mathcal{R}^\star$ be the optimizer of problem* (1) *and each agent $i \in \mathcal{A}$ follow the update rules* (9)*, and* (10) *with the initial condition of $\mathbf{x}_i(0) = \mathbf{0}$. Then,*

$$\left(1-\frac{1}{\mathrm{e}}\right)\left(1-\left(2N^2d(\mathcal{G})+\frac{1}{2}N^2+N\right)\frac{1}{T}\right)f(\mathcal{R}^\star) \leq F(\bar{\mathbf{x}}(T)),$$

holds with probability of at least $1-2T\,n\,\mathrm{e}^{-\frac{1}{8T^2}\underline{K}}$, $\underline{K} = \min_{i\in\mathcal{A}} K_i$.

Proof of Lemma 3.2 is given in the extended version of this paper available at [19].

$\bar{\mathbf{x}}(T)$, however, is a fractional point in $\mathcal{M}$. Moreover, only part of it is available at each agent $i \in \mathcal{A}$. The final output of a distributed solver for problem (1) must be a set $\bar{\mathcal{R}}$ that should belong to the constraint set $\mathcal{I}$ in (1b). From the propagation and update rules (9) and (10) and Lemma 3.1, we can conclude that $\mathbf{1}.\mathbf{x}_{ii}(T) = 1$. We propose that each agent $i \in \mathcal{A}$ samples a single policy $\bar{p}_i$ out of its local policy set $\mathcal{P}_i$ according to probability membership vector $\mathbf{x}_{ii}(T)$ (according to Algorithm 2, whose notation is introduced in Section III-C), which results in the distributively selected final policy set $\bar{\mathcal{R}} = \bigcup_{i\in\mathcal{A}}\{\bar{p}_i\}$. The following lemma, whose proof is given in the extended version of this paper available at [19], compares the utility value $f(\bar{R})$ with $F(\bar{x}(T))$ in the expected value.

Lemma 3.3 (Distributed rounding) *Letting each agents $i \in \mathcal{A}$ to sample one single policy $\bar{p}_i$ out of $\mathcal{P}_i$ according to the probability membership vector $\mathbf{x}_{ii}(T)$ results in $\bar{\mathcal{R}} = \bigcup_{i\in\mathcal{A}}\{\bar{p}_i\}$ that satisfies the constraint* (1b) *and*

$$F(\bar{\mathbf{x}}(T)) \leq \mathbb{E}[f(\bar{\mathcal{R}})].$$

Algorithm 1 Distributed multilinear-extension-based iterative greedy algorithm

```
1: Init: $\bar{\mathcal{P}} \leftarrow \emptyset$, $\mathcal{F}_i \leftarrow \emptyset$, $t \leftarrow 1$,
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 probability $\alpha$ for all $(q,\alpha) \in \mathcal{F}_i$.
5:         for $p \in \mathcal{P}_i$ do
6:             Compute $w^i_p \sim \mathbb{E}[f(\mathcal{R}\cup\{p\}) - f(\mathcal{R}\setminus\{p\})]$ using the policy sample sets of step 4.
7:         end for
8:         Solve for $p^\star = \underset{p\in\mathcal{P}_i}{\operatorname{argmax}}\ w^i_p$.
9:         $\mathcal{F}^-_i \leftarrow \mathcal{F}_i \oplus \{(p^\star, \frac{1}{T})\}$.
10:        Broadcast $\mathcal{F}^-_i$ to the neighbors $\mathcal{N}_i$.
11:        $\mathcal{F}_i \leftarrow \underset{j\in\mathcal{N}_i\cup\{i\}}{\text{MAX}}\ \mathcal{F}^-_j$
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 $\mathcal{F}_i$ using Algorithm 2
17:     $\bar{\mathcal{P}} \leftarrow \bar{\mathcal{P}} \cup \{\bar{p}\}$
18: end for
```

Algorithm 2 Distributed Policy Selection

```
1: Input: $\mathcal{P}_i, \mathcal{F}_i$.
2: Let $\mathcal{P}_i(l)$, $l \in \{1,\cdots,|\mathcal{P}_i|\}$ be the $l$th element of $\mathcal{P}_i$.
3: Generate vector $\mathbf{y}_i \in \mathbb{R}^{|\mathcal{P}_i|}_{\geq 0}$ as follows. For any $\mathcal{P}_i(l) \in \mathcal{P}_i$, $l \in \{1,\cdots,|\mathcal{P}_i|\}$,
   let $\begin{cases}[\mathbf{y}_i]_l = \alpha, & (\mathcal{P}_i(l),\alpha) \in \mathcal{F}_i(T),\\ [\mathbf{y}_i]_l = 0, & \text{otherwise}.\end{cases}$
4: Generate a random number $\beta \in [0,1]$ with a uniform distribution.
5: Find a $\bar{p} = \mathcal{P}_i(l) \in \mathcal{P}_i$, $l \in \{1,\cdots,|\mathcal{P}_i|\}$, such that $\sum_{k=1}^{l}[\mathbf{y}_i]_k \leq \beta \leq \sum_{k=1}^{l+1}[\mathbf{y}_i]_k$
6: return $\bar{p}$
```

Note: notice that $\mathcal{F}_i$ here is the information set of agent $i \in \mathcal{A}$ at $t = T$.

C. Distributed multilinear-extension-based iterative greedy algorithm: A minimal information implementation

In what follows, we outline how the propagation and update rules (9) and (10) can be implemented in a distributed way without a need to exchange the entire $\mathbf{x}_i \in \mathbb{R}^n$ of each agent $i \in \mathcal{A}$ with its neighbors. The resulted implementation is summarized as the distributed multilinear-extension-based iterative greedy algorithm presented as Algorithm 1.

We define the local information set of each agent i at time t as

$$\mathcal{F}_i(t) = \left\{(p,\alpha) \in \mathcal{P}\times[0,1] \,\middle|\, [\mathbf{x}_i(t)]_p \neq 0 \text{ and } \alpha = [\mathbf{x}_i(t)]_p\right\}. \tag{13}$$

Since $\mathbf{x}^i(0) = \mathbf{0}$, then $\mathcal{F}_i(0) = \{\}$. Introduction of the information set $\mathcal{F}_i(t)$ provides the framework through which the agents only store and communicate the necessary information. Furthermore, it enables the agents to evaluate the necessary values only using the available information in $\mathcal{F}_i(t)$.

It follows from f being monotone increasing and submodular that $f(\mathcal{R}_{\mathbf{x}_i(t)}\cup\{p\}) - f(\mathcal{R}_{\mathbf{x}_i(t)}\setminus\{p\}) \geq 0$. Therefore, one realization of $\widetilde{\mathbf{v}}_i(t)$ of problem (8) is $\mathbf{1}_{\{p^\star_i\}}$, where

$$p^\star_i = \underset{p\in\mathcal{P}_i}{\arg\max}\ w^i_p(t), \tag{14}$$

where agent i can empirically evaluate $w^i_p(t)$ using the local information set $\mathcal{F}_i(t)$ as

$$w_p^i(t) = \mathbb{E}[f(R \cup \{p\}) - f(R \setminus \{p\}]$$

for all $p \in \mathcal{P}_i$ by generating K_i samples of R such that $p \in \mathcal{R}$ with the probability of α for all couples $(p, \alpha) \in \mathcal{F}_i(t)$.

Since $\mathbf{1}_{\{p_i^\star\}}$ is a realization of $\widetilde{\mathbf{v}}_i(t)$, the corresponding realization of update rule (9) over the information set $\mathcal{F}_i(t)$ is denoted as

$$\mathcal{F}_i^-(t+1) = \mathcal{F}_i(t) \oplus \left\{ (p_i^\star, \frac{1}{T}) \right\}, \tag{15}$$

where $\mathcal{F}_i^-(t+1)$ is consistent with the realization of $\mathbf{x}_i^-(t+1)$ through the membership probability vector to information set conversion relation (13).

Instead of the agents sharing $\mathbf{x}_i^-(t)$, $i \in \mathcal{A}$ with their neighbors, they can share their local information set with their neighboring agents and execute a max operation over their local and received information sets as

$$\mathcal{F}_i(t+1) = \underset{j \in \mathcal{N}_i \cup \{i\}}{\text{MAX}} \mathcal{F}_j^-(t+1). \tag{16}$$

Consequently, through the membership probability vector to information set conversion relation (13) $\mathcal{F}_i(t+1)$ is consistent with a realization of $\mathbf{x}_i(t+1)$.

Finally given the information set of each agent $i \in \mathcal{A}$ at time T, each agents samples a single policy $\bar{p}_i \in \mathcal{P}_i$ such that

$$\bar{p}_i = p, \quad \text{with probability} \quad \alpha, \tag{17}$$

for all $(p, \alpha) \in \mathcal{F}_i(T)$ and $p \in \mathcal{P}_i$.

The following theorem establishes the optimality bound of $f(\bar{\mathcal{R}})$ where $\bar{\mathcal{R}} = \bigcup_{i \in \mathcal{A}} \{\bar{p}_i\}$ is generated through the decentralized Algorithm 1.

Theorem 3.1 (Convergence guarantee and suboptimality gap of Algorithm 1) *Let $f : 2^{\mathcal{P}} \to \mathbb{R}_{\geq 0}$ be normalized, monotone increasing and submodular set function. Let $\mathcal{R}^\star$ to be the optimizer of problem* (1)*. Following the distributed Algorithm 1, the admissible policy set $\bar{\mathcal{P}}$ is achieved such that*

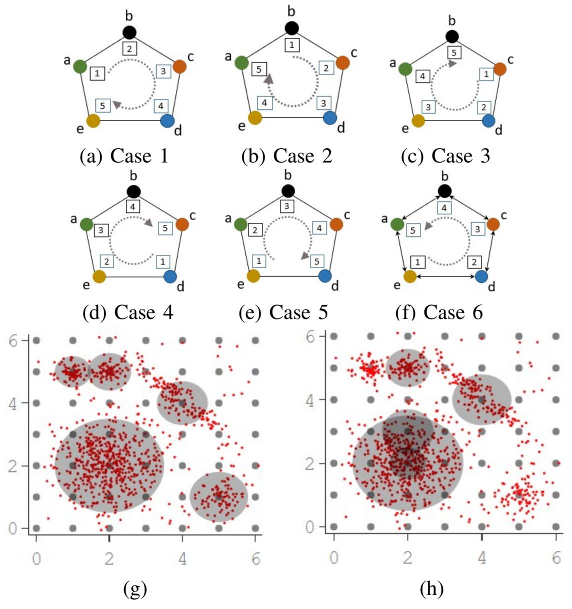
$$\left(1 - \frac{1}{\mathrm{e}}\right)\left(1 - \left(2N^2 d(\mathcal{G}) + \frac{1}{2}N^2 + N\right)\frac{1}{T}\right) f(\mathcal{R}^\star) \leq \mathbb{E}[f(\bar{\mathcal{R}})],$$

holds with probability of at least $1 - 2T\,n\,\mathrm{e}^{-\frac{1}{8T^2}\underline{K}}$, $\underline{K} = \min_{i \in \mathcal{A}} K_i$.

Proof: Given that the information set update rules (14), (15), and , (16) are a realization of the vector space update rules rules (8), (9), and (10), we can conclude that the vector $\mathbf{y} = [\mathbf{y}_1^\top, \cdots, \mathbf{y}_N^\top]^\top$ defined as

$$\begin{cases} [\mathbf{y}]_p = \alpha, & (p, \alpha) \in \mathcal{F}_i(T), \quad p \in \mathcal{P}_i \\ [\mathbf{y}]_p = 0, & \text{Otherwise} \end{cases}$$

is a realization of $\bar{\mathbf{x}}(T)$ and satisfies $F(\bar{\mathbf{x}}(T)) = F(\mathbf{y})$. Moreover, sampling a single policy $\bar{p}_i$ according to $\mathbf{y}_i$ out of $\mathcal{P}_i$ is equivalent to sampling rule (17). Noting that $\mathbf{y}$ is a realization of $\bar{\mathbf{x}}(T)$, Lemma 3.2 and Lemma 3.3 leads us to concluding the proof. ■



(a) Case 1 (b) Case 2 (c) Case 3
(d) Case 4 (e) Case 5 (f) Case 6
(g) (h)

Fig. 2: Plots (a)-(f) show 6 different SEQ used in the sequential greedy algorithm. Plot (g) shows the outcome of using Algorithm 1 whereas plot (h) shows the outcome of the sequential greedy algorithm when SEQ in case 1 (plot (a)) is used.

IV. Numerical Example

Consider the multi-agent sensor placement problem introduced in Fig. 1 for 5 agents for which $\mathcal{B}_i = \mathcal{B}$, i.e., each agent can move to any of the sensor placement points. This sensor placement point is cast by optimization problem (1). 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 $\mathcal{A} = \{a, b, c, d, e\}$ 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 proposed distributed Algorithm 1. The results of the simulation for different iteration and sampling numbers are shown in Table I. The algorithm produces good results at the 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 1. Next, we solve this algorithm using the sequential greedy algorithm [1] in a decentralized way by first choosing a route SEQ $= \boxed{1} \rightarrow \boxed{2} \rightarrow \boxed{3} \rightarrow \boxed{4} \rightarrow \boxed{5}$ that visits all the agents, and then giving SEQ to the agents so they follow SEQ to share their information in a sequential manner. Fig. 2(a)-(f) gives 6 possible SEQ denoted by the semi-circular arrow inside the networks. The results of running the sequential greedy algorithm over the sequences in Fig. 2(a)-(f) are shown in Table II. What stands out about the sequential greedy algorithm is that the choice of sequence can significantly affect the algorithm's outcome. 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

T \ K	10000	500	100	50	10	5	1
100	768	768	718	710	718	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

TABLE I: The outcome of Algorithm 1 for different iteration T and sampling numbers $\underline{K}$.

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

TABLE II: Outcome of sequential greedy algorithm.

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. 2(h), which depicts the outcome of the sequential greedy algorithm using the sequence in Case 1. A simpler example justifying this assumption is shown in Fig. 3 with the two disjoint clusters of points and two sensors. One may suggest sequencing the agents from high to low sensing zone order; however, this is not necessarily the best choice as we can see in Table II, 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 1 where the agents intrinsically are overcoming the importance of a sequence by deciding the position of the sensor over a time horizon of communication and exchanging their information set.

V. Conclusion

We proposed a distributed suboptimal algorithm to solve the problem of maximizing an 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 $(1-1/\mathrm{e}-O(1/T))$ optimality 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 optimality gap over the sequential greedy algorithm. Our future work is to study the robustness of our proposed algorithm to message dropout.

References

[1] N. Rezazadeh and S. S. Kia, "A sub-modular receding horizon solution for mobile multi-agent persistent monitoring," *Automatica*, vol. 127, p. 109460, 2021.

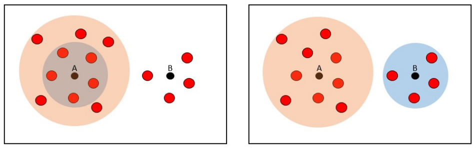


Fig. 3: In the sequential greedy algorithm, when the blue agent chooses first, it assigns both the blue and the orange agents to point A resulting in an inferior performance compared to the case that the orange agent chooses first. In the later case, orange agent gets point A and the blue agent gets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

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

[3] 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.

[4] C. Chekuri and A. Kumar, "Maximum coverage problem with group budget constraints and applications," in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pp. 72–83, Springer, 2004.

[5] 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.

[6] 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.

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

[8] 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.

[9] 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.

[10] 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.

[11] P. S. Raut, O. Sadeghi, and M. Fazel, "Online dr-submodular maximization with stochastic cumulative constraints," *arXiv preprint arXiv:2005.14708*, 2020.

[12] J. Vondrák, "Optimal approximation for the submodular welfare problem in the value oracle model," in *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pp. 67–74, 2008.

[13] A. A. Bian, B. Mirzasoleiman, J. Buhmann, and A. Krause, "Guaranteed non-convex optimization: Submodular maximization over continuous domains," in *Artificial Intelligence and Statistics*, pp. 111–120, 2017.

[14] A. Mokhtari, H. Hassani, and A. Karbasi, "Stochastic conditional gradient methods: From convex minimization to submodular maximization," *Journal of Machine Learning Research*, vol. 21, no. 105, pp. 1–49, 2020.

[15] O. Sadeghi and M. Fazel, "Online continuous dr-submodular maximization with long-term budget constraints," in *International Conference on Artificial Intelligence and Statistics*, pp. 4410–4419, 2020.

[16] A. Robey, A. Adibi, B. Schlotfeldt, J. G. Pappas, and H. Hassani, "Optimal algorithms for submodular maximization with distributed constraints," *arXiv preprint arXiv:1909.13676*, 2019.

[17] A. A. Ageev and M. I. Sviridenko, "Pipage rounding: A new method of constructing algorithms with proven performance guarantee," *Journal of Combinatorial Optimization*, vol. 8, no. 3, pp. 307–328, 2004.

[18] H. W, "Probability inequalities for sums of bounded random variables," in *The Collected Works of Wassily Hoeffding*, pp. 409–426, Springer, 1994.

[19] N. Rezazadeh and S. S. Kia, "Multi-agent maximization of a monotone submodular function via maximum consensus," 2020.