Decentralized Gradient Tracking for Continuous DR-Submodular Maximization

Jiahao Xie **Chao Zhang** **Zebang Shen** **Chao Mi** **Hui Qian***
College of Computer Science and Technology, Zhejiang University, China
{xiejh, zczju, shenzebang, michao, qianhui}@zju.edu.cn

Abstract

In this paper, we focus on the continuous DR-submodular maximization over a network. By using the gradient tracking technique, two decentralized algorithms are proposed for deterministic and stochastic settings, respectively. The proposed methods attain the ϵ-accuracy tight approximation ratio for monotone continuous DR-submodular functions in only $\mathcal{O}(1/\epsilon)$ and $\tilde{\mathcal{O}}(1/\epsilon)$ rounds of communication, respectively, which are superior to the state-of-the-art. Our numerical results show that the proposed methods outperform existing decentralized methods in terms of both computation and communication complexity.

1 Introduction

The algorithms explored in this paper aim to find an approximate solution to a continuous *DR-submodular* maximization problem in a *decentralized* and *consensus* setting, where each node merely has access to a subset of data and are allowed to exchange information with their neighboring nodes only. Specifically, the problem of interest can be phrased as

$$\max_{\boldsymbol{x}\in\mathcal{C}} F(\boldsymbol{x}) = \max_{\substack{\boldsymbol{x}_1=\cdots=\boldsymbol{x}_n \\ \boldsymbol{x}_i\in\mathcal{C}}} \Big\{\frac{1}{n}\sum_{i=1}^{n} F_i(\boldsymbol{x}_i)\Big\}, \qquad (1)$$

where F_i is a local monotone continuous DR-submodular function, and $\mathcal{C}$ is a compact convex set.

In such a regime, we say that an algorithm achieves a γ-approximation ratio ($\gamma \in (0,1]$) if it finds a solution $\hat{\boldsymbol{x}}$ such that $F(\hat{\boldsymbol{x}}) \geq \gamma F(\boldsymbol{x}^*)$, where $\boldsymbol{x}^*$ denote the global maximizer of F. Throughout this paper, we use *tight* approximation ratio to refer to the $(1-1/e)$ approximation ratio and the ϵ-accuracy tight approximation ratio to refer to the $(1-1/e-\epsilon)$ approximation ratio unless otherwise specified[1].

Research on problem (1) has received an enormous impetus from the maturity of the research on continuous DR-submodular functions, a broad subclass of nonconvex functions with diminishing returns (DR) property. Various applications, including the design of online experiments (Chen et al., 2018), budget and resource allocations (Eghbali and Fazel, 2016; Staib and Jegelka, 2017), and learning assignments (Golovin et al., 2014) are captured in this regime. However, most existing work suffers from centralized computing (single-machine setting) in that large-scale submodular maximization involves many information gathering, data summarization, and non-parametric learning problems. This necessitates the development of distributed methods for submodular maximization. Besides, problem (1) also arises in tasks such as sensor networks (Golovin et al., 2010) and multirobot systems (Singh et al., 2007) where either efficiently centralizing data or globally aggregating intermediate results is unfeasible.

An alternative distributed setting for continuous DR-submodular maximization relies on a control (master) node that exchanges information with all the other computing agents (workers). Usually, such setting is not robust to machine failures or network topological changes. Moreover, when the network is large-scale and sparse, master-worker distributed methods usually have high communication time (Mokhtari et al., 2018b). In contrast, the decentralized distributed setting only relies on local communication between neighboring nodes, and thus have advantages of robustness to machine and link failures, scalability to network sizes, and privacy preservation.

*Corresponding author.

Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89.

[1] It is proved that the approximation guarantee can be tightened to $(1-1/e)$ by using Frank-Wolfe(Bian et al., 2017), or stochastic Frank-Wolfe(Mokhtari et al., 2018a)

Recently, Mokhtari et al. (2018b) propose two decentralized Frank-Wolfe variants named DeCG and DeSCG[2]for maximizing monotone continuous DR-submodular functions, where DeCG is deterministic as it requires *full local gradient* evaluations at each node while DeSCG is stochastic in that only an unbiased gradient estimator is calculated for each node. In these two methods, each node maintains a *surrogate* for the global gradient estimated from both local and neighboring gradient information.

The first of their kind, DeCG and DeSCG have two shortcomings. The **first** is that the computation complexity of DeCG mismatches $\mathcal{O}(1/\epsilon)$, the best computation complexity of deterministic Frank-Wolfe type methods (Bian et al., 2017). Specifically, DeCG constructs a surrogate for the global gradient at each node by aggregating the local gradient and neighboring gradient information without effectively using historical local gradient information. Such surrogate does not provide good approximation of the global gradient by design, which results in $\mathcal{O}(1/\epsilon^2)$ iteration/computation complexity of the algorithm. The **second** shortcoming is the high communication overhead. Although both methods achieve the tight approximation ratio, their communication complexities are rather high because (i) the surrogate adopted by both methods has high global gradient approximation error which dominates the convergence rate and thus results in high iteration complexity or equivalently communication complexity; (ii) DeSCG uses a fixed mini-batch size in constructing local gradient estimators, which leads to low per-iteration computation cost but sacrifices the accuracy of the estimators and increases the overall communication complexity to reach convergence. They overlooked such practical issue, which restrains their applications.

To bridge these gaps, we adopt the gradient tracking technique (Di Lorenzo and Scutari, 2016; Wai et al., 2017; Pu and Nedić, 2018) to construct decentralized Frank-Wolfe variants, DeGTFW and DESGTFW, for deterministic and stochastic settings, respectively. Both algorithms can be used to efficiently maximize monotone continuous DR-submodular functions subject to any compact convex set. Our main contributions are listed as follows.

1. We prove that DeGTFW achieves the tight approximation ratio for maximizing monotone, continuous-DR submodular functions subject to any compact convex set. Both the communication and gradient evaluation complexities of DeGTFW are $\mathcal{O}(1/\epsilon)$ to achieve an ϵ-accuracy tight $(1-1/e-\epsilon)$ approximation ratio, while these complexities of DeCG are both $\mathcal{O}(1/\epsilon^2)$.
2. Compared to $\mathcal{O}(1/\epsilon^3)$ communication and gradient evaluation complexities of DeSCG, DeSGTFW requires $\tilde{\mathcal{O}}(1/\epsilon)$ communication complexity[3]and $\tilde{\mathcal{O}}(1/\epsilon^3)$ stochastic gradient evaluation complexity to achieve an ϵ-accuracy tight approximation ratio in expectation.

We conduct numerical experiments on Non-convex / non-concave Quadratic Programming (NQP) and multilinear extension. The empirical results show that the proposed methods outperform existing decentralized methods in terms of both computation complexity and communication complexity.

2 Related Work

Submodularity, a structural property of functions in both discrete and continuous domains, captures a wide range of real-world applications including document summarization (Lin and Bilmes, 2010), recommender systems (El-Arini et al., 2009; Yue and Guestrin, 2011), active learning (Golovin and Krause, 2011), etc. For many of these applications, one important task is to find the global maximum of a submodular function. Although exact maximization of submodular functions is NP-hard (Feige, 1998; Bian et al., 2017), they can be approximately maximized in polynomial time (Krause and Golovin, 2014). Existing literature of submodular maximization can be divided to two regimes: the submodular set function maximization (discrete domain) and the continuous submodular maximization. We also review the existing work for distributed submodular maximization.

Submodular Set Function Maximization. The celebrated work of (Nemhauser et al., 1978) proposes a discrete greedy algorithm that achieves the tight approximation ratio for maximizing a monotone submodular set function subject to a cardinality constraint. Following this line of work, various greedy algorithms have been proposed for more complex constraints such as knapsack constraints (Wolsey, 1982; Sviridenko, 2004), linear packing constraints (Azar and Gamzu, 2012), and matroid constraints (Krause et al., 2009; Stan et al., 2017).

Continuous Submodular Maximization. The continuous submodular maximization problem arises in many applications, such as submodular NQP, generalized facility location, optimal budget allocation,

[2]The second method in their paper is in fact a discrete variant of DeCG, which is essentially a stochastic variant of DeCG with an additional rounding step. We refer to the stochastic variant without the rounding step as DeSCG for ease of statement.

[3]where $\tilde{\mathcal{O}}()$ suppresses a poly-logarithmic factor.

text summarization, to name a few (see (Bian et al., 2017)). Another important collection of continuous submodular functions originates from the multilinear extension of discrete submodular set functions. In the seminal work of (Vondrák, 2008), the authors introduce the multilinear extension technique for lifting a monotone submodular set function with a matroid constraint to continuous domains. A continuous-time algorithm with the tight approximation ratio, named Continuous Greedy (CG), is proposed to solve the corresponding continuous DR-submodular maximization problem. The authors also show that by using the pipage rounding technique (Ageev and Sviridenko, 2004), the fractional solution obtained by continuous DR-submodular maximization can be rounded to a feasible discrete solution without loss in objective value, which achieves the tight approximation ratio for the original discrete problem.

Recently, Bian et al. (2017) propose a generalized version of CG which attains an ϵ-accuracy tight approximation ratio for continuous monotone DR-submodular maximization under down-closed bounded convex constraints with $\mathcal{O}(1/\epsilon)$ gradient evaluation complexity. However, when applied to the stochastic setting, their algorithm can produce solutions that are arbitrarily bad. To tackle this problem, Hassani et al. (2017) propose a Stochastic Projected Gradient (SPG) method that attains a $(1/2-\epsilon)$ approximation ratio for stochastic continuous monotone DR-submodular maximization under general convex constraints with $\mathcal{O}(1/\epsilon^2)$ stochastic gradient evaluation complexity. The authors also show that $1/2$ is the best approximation ratio one can obtain for projected gradient methods, indicating a gap to the tight approximation ratio. In order to close this gap in the stochastic setting, Mokhtari et al. (2018a) propose a stochastic method named Stochastic Continuous Greedy (SCG) which achieves an ϵ-accuracy tight approximation ratio with a stochastic gradient evaluation complexity of $\mathcal{O}(1/\epsilon^3)$.

Distributed submodular maximization. Unlike methods in the single-machine setting, distributed methods focus on solving a global problem distributed over a set of computing agents connected by a network. Existing algorithms for distributed submodular maximization include master-worker algorithms and decentralized algorithms.

Several master-worker algorithms for discrete submodular maximization have been proposed (Golovin et al., 2010; Mirzasoleiman et al., 2013; Barbosa et al., 2015; Kumar et al., 2015). In these algorithms, each machine finds a local solution using its local data and then sends the solution to a master node which computes the global solution by aggregating all local solutions.

Decentralized methods only require local communication between neighboring nodes in the network. Gharesifard and Smith (2016) propose a decentralized algorithm for a special discrete submodular maximization problem subject to a partition matroid constraint. In their method, the nodes take actions in a predefined sequential order based on information of preceding nodes in the neighborhood. However, the approximation ratio of such algorithm is not tight and the computation over the network cannot be parallelized. Mokhtari et al. (2018b) are the first to consider decentralized continuous DR-submodular maximization. They propose a deterministic method named DeCG that achieves an ϵ-accuracy tight approximation ratio for monotone continuous DR-submodular maximization problems subject to any down-closed bounded convex set. The gradient evaluation and communication complexities of DeCG are both $\mathcal{O}(1/\epsilon^2)$. For the stochastic setting, the authors further present a variant called DeSCG which achieves an ϵ-accuracy tight approximation ratio in expectation with $\mathcal{O}(1/\epsilon^3)$ stochastic gradient evaluation and communication complexities. We summarize all related algorithms in Table 1.

3 Notations and Preliminaries

We use bold lowercase symbols to denote vectors and bold uppercase symbols to denote matrices. The i-th row of matrix $\boldsymbol{W}$ is represented by $\boldsymbol{W}_{i*}$. For two vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$, the notation $\boldsymbol{x} < \boldsymbol{y}$ means that $\boldsymbol{x}$ is component-wise less than $\boldsymbol{y}$. Throughout this paper, we let $\|\boldsymbol{x}\|$ denote the Euclidean norm of a vector $\boldsymbol{x}$ and $\|\boldsymbol{W}\|$ denote the spectral norm of a matrix $\boldsymbol{W}$. The kronecker product of matrices is denoted by $\otimes$. The notation $[d]$ stands for the set $\{1, \ldots, d\}$. We let $\boldsymbol{e}_j$ denote a basis vector in $\mathbb{R}^d$ where the j-th entry is 1, and all the others are 0. We use $\mathbf{1}_d \in \mathbb{R}^d$ to denote a vector with all d components being 1 and $\mathbf{0}_d \in \mathbb{R}^d$ to denote a vector with all d components being 0. The identity matrix in $\mathbb{R}^{d\times d}$ is represented by $\boldsymbol{I}_d$.

In what follows, we present definitions of submodularity, DR-submodularity, and monotonicity, respectively, and describe our network model precisely.

Submodularity. Consider a continuous function $F_i : \mathcal{X} \to \mathbb{R}_+$, where $\mathcal{X} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_d \in \mathbb{R}_+^d$ and $\mathcal{X}_j \subset \mathbb{R}_+$ be a closed sub-interval for each $j \in [d]$, F_i is called submodular if for any $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{X} \times \mathcal{X}$,

$$F_i(\boldsymbol{x}) + F_i(\boldsymbol{y}) \geq F_i(\boldsymbol{x} \vee \boldsymbol{y}) + F_i(\boldsymbol{x} \wedge \boldsymbol{y}), \qquad (2)$$

where $\vee$ and $\wedge$ are component-wise maximum and minimum, respectively.

DR-submodularity. A differentiable function F_i is

Table 1: Summary of CG, SG, SCG, DeCG, DeSCG, DeGTFW, and DeSGTFW.

Algorithm	CG	SPG	SCG	DeCG	DeSCG	DeGTFW	DeSGTFW
setting	det.	stoch.	stoch.	det.	stoch.	det.	stoch.
constraint	cvx-down	convex	convex	cvx-down	cvx-down	convex	convex
approx. ratio	$1-1/e-\epsilon$	$1/2-\epsilon$	$1-1/e-\epsilon$	$1-1/e-\epsilon$	$1-1/e-\epsilon$	$1-1/e-\epsilon$	$1-1/e-\epsilon$
computation	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(1/\epsilon^2)$	$\mathcal{O}(1/\epsilon^3)$	$\mathcal{O}(1/\epsilon^2)$	$\mathcal{O}(1/\epsilon^3)$	$\mathcal{O}(1/\epsilon)$	$\tilde{\mathcal{O}}(1/\epsilon^3)$
communication	\	\	\	$\mathcal{O}(1/\epsilon^2)$	$\mathcal{O}(1/\epsilon^3)$	$\mathcal{O}(1/\epsilon)$	$\tilde{\mathcal{O}}(1/\epsilon)$

called DR-submodular if it exhibits diminishing returns, i.e., for any $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}$ and $\boldsymbol{x} \leq \boldsymbol{y}$,

$$\nabla F_i(\boldsymbol{x}) \geq \nabla F_i(\boldsymbol{y}). \tag{3}$$

Monotonicity. A function F_i is called monotone if

$$F_i(\boldsymbol{x}) \leq F_i(\boldsymbol{y}),\ \forall\ \boldsymbol{x}, \boldsymbol{y} \in \mathcal{X} \text{ and } \boldsymbol{x} \leq \boldsymbol{y}. \tag{4}$$

Without loss of generality, we assume that $\mathcal{X} = \{\boldsymbol{x} | \mathbf{0}_d \leq \boldsymbol{x} \leq \boldsymbol{u}\}$. This is because for any monotone continuous DR-submodular function F_i with a domain $\mathcal{X} = \{\boldsymbol{x} | \boldsymbol{\ell} \leq \boldsymbol{x} \leq \boldsymbol{u}\}$, one can define a new domain $\mathcal{X}' = \{\boldsymbol{x} | \mathbf{0}_d \leq \boldsymbol{x} \leq \boldsymbol{u} - \boldsymbol{\ell}\}$ and transform F_i to $F_i'(\boldsymbol{x}) = F_i(\boldsymbol{x} + \boldsymbol{\ell})$. Besides, the monotonicity of F_i implies that $\nabla F_i(\boldsymbol{x}) \geq \mathbf{0}_d$ for any $\boldsymbol{x} \in \mathcal{X}$. Therefore, for any $\boldsymbol{x} \in \mathcal{X}$,

$$\mathbf{0}_d \leq \nabla F(\boldsymbol{x}) \leq \nabla F_i(\mathbf{0}_d). \tag{5}$$

Network Model. The network model we consider in this paper is represented by an undirected connected graph $\mathcal{G} = (V, E)$ with n nodes. Each node $i \in [n]$ represents a computing agent and is associated with a local objective function F_i. We denote the set of neighbors of node i by $\mathcal{N}(i) = \{j | (i, j) \in E\}$.

4 DeGTFW and DeSGTFW Algorithms

In this section, we propose two decentralized methods for solving problem (1). The first method is a deterministic method named Decentralized Gradient Tracking Frank-Wolfe (DeGTFW) that requires evaluation of full local gradients ∇F_i. The second method is a stochastic one named Decentralized Stochastic Gradient Tracking Frank-Wolfe (DeSGTFW) which utilizes an unbiased estimator of full local gradients.

In both methods, an auxiliary weight matrix $\boldsymbol{W}$ is constructed to aggregate received information from neighbors, where $\boldsymbol{W}_{ij}$ denotes the weight node i assigns to node j. The weight matrix $\boldsymbol{W}$ depends on the network topology and should be chosen properly to ensure that the local solutions found by individual nodes reach consensus. Specifically, $\boldsymbol{W} \in \mathbb{R}_+^{n \times n}$ should satisfy the following assumption (Yuan et al., 2016).

Assumption 1. *Each entry of the weight matrix $\boldsymbol{W}_{ij} > 0$ if $j \in \mathcal{N}(i) \cup \{i\}$, and $\boldsymbol{W}_{ij} = 0$ otherwise. Besides, $\boldsymbol{W}$ is symmetric and doubly stochastic (i.e., $\boldsymbol{W}\mathbf{1}_n = \boldsymbol{W}^\top \mathbf{1}_n = \mathbf{1}_n$). Further, its second largest (in magnitude) eigenvalue is strictly smaller than 1, i.e.,*

$$\beta := \max(|\lambda_2(\boldsymbol{W})|, |\lambda_n(\boldsymbol{W})|) < 1, \tag{6}$$

where $\lambda_i(\boldsymbol{W})$ denotes the i-th largest eigenvalue of $\boldsymbol{W}$, i.e., $1 = \lambda_1(\boldsymbol{W}) > \lambda_2(\boldsymbol{W}) \geq \ldots \geq \lambda_n(\boldsymbol{W})$.

4.1 Decentralized Gradient Tracking Frank-Wolfe

Now we present the DeGTFW algorithm, which is summarized in Algorithm 1. In this algorithm, each node maintains a local variable $\boldsymbol{x}_i^{(t)}$, a local gradient substitute $\boldsymbol{g}_i^{(t)}$ and a global gradient surrogate $\boldsymbol{d}_i^{(t)}$, where the superscript (t) denotes the iteration index and the subscript i denotes the node index. At each iteration, each node i obtains $\boldsymbol{x}_j^{(t)}$ and $\boldsymbol{g}_j^{(t)}$ from all its neighbors and then computes the global gradient surrogate $\boldsymbol{d}_i^{(t)}$ as the weighted average of $\boldsymbol{g}_j^{(t)}$ for $j \in \mathcal{N}(i) \cup \{i\}$. Then an ascent direction is found by solving the linear program $\boldsymbol{v}_i^{(t)} = \operatorname{argmax}_{\boldsymbol{v} \in \mathcal{C}} \langle \boldsymbol{v}, \boldsymbol{d}_i^{(t)} \rangle$. After that, the local variable $\boldsymbol{x}_i^{(t)}$ is updated by aggregating the received $\boldsymbol{x}_j^{(t)}$ from neighbors and taking a small step towards the ascent direction $\boldsymbol{v}_i^{(t)}$,

$$\boldsymbol{x}_i^{(t+1)} = \sum_{j \in \mathcal{N}(i) \cup \{i\}} \boldsymbol{W}_{ij} \cdot \boldsymbol{x}_j^{(t)} + \frac{1}{T} \boldsymbol{v}_i^{(t)}. \tag{7}$$

Finally, the local gradient substitute $\boldsymbol{g}_i^{(t)}$ is updated as

$$\boldsymbol{g}_i^{(t+1)} = \boldsymbol{d}_i^{(t)} + \nabla F_i(\boldsymbol{x}_i^{(t+1)}) - \nabla F_i(\boldsymbol{x}_i^{(t)}), \tag{8}$$

where $\nabla F_i(\boldsymbol{x}_i^{(t)})$ is the old local gradient evaluated at the last iteration.

The substitute (8) is analogous to that used in the variance reduced stochastic gradient method SAGA (Defazio et al., 2014) which leverages history stochastic gradient information to approximate full gradients. Actually this gradient tracking technique has been adopted by decentralized methods for convex and

Algorithm 1: Decentralized Gradient Tracking Frank-Wolfe for node i

Input : weight matrix $\boldsymbol{W} \in \mathbb{R}_+^{n\times n}$, constraint set $\mathcal{C} \subset \mathbb{R}_+^n$, number of iterations T

Initialize $\boldsymbol{x}_i^{(1)} = 0$, $\boldsymbol{g}_i^{(1)} = \nabla F_i(\boldsymbol{x}_i^{(1)})$

for $t = 1, 2, \ldots, T$ **do**

Obtain $\boldsymbol{g}_j^{(t)}$ and $\boldsymbol{x}_j^{(t)}$ from neighbors $j \in \mathcal{N}(i)$;

$\boldsymbol{d}_i^{(t)} = \sum_{j\in\mathcal{N}(i)\cup\{i\}} \boldsymbol{W}_{ij} \cdot \boldsymbol{g}_j^{(t)}$;

$\boldsymbol{v}_i^{(t)} = \operatorname{argmax}_{\boldsymbol{v}\in\mathcal{C}} \langle \boldsymbol{v}, \boldsymbol{d}_i^{(t)} \rangle$;

$\boldsymbol{x}_i^{(t+1)} = \sum_{j\in\mathcal{N}(i)\cup\{i\}} \boldsymbol{W}_{ij} \cdot \boldsymbol{x}_j^{(t)} + \frac{1}{T}\boldsymbol{v}_i^{(t)}$;

$\boldsymbol{g}_i^{(t+1)} = \boldsymbol{d}_i^{(t)} + \nabla F_i(\boldsymbol{x}_i^{(t+1)}) - \nabla F_i(\boldsymbol{x}_i^{(t)})$

end

Return: $\boldsymbol{x}_i^{(T+1)}$

general non-convex optimization problems (Di Lorenzo and Scutari, 2016; Wai et al., 2017; Qu and Li, 2017; Pu and Nedić, 2018). Although continuous submodular functions are a subclass of non-convex/non-concave functions, most existing decentralized methods for general non-convex problems only guarantee to find a stationary point which, unfortunately, does not provide the tight approximation ratio for continuous DR-submodular maximization (Hassani et al., 2017). On the contrary, Algorithm 1 is guaranteed to achieve an ϵ-accuracy tight approximation ratio with an iteration/communication complexity of $\mathcal{O}(1/\epsilon)$. In comparison to the global gradient surrogate we use in DeGTFW, Mokhtari et al. (2018b) propose to use another surrogate

$$\boldsymbol{d}_i^{(t)} = (1-\alpha) \sum_{j\in\mathcal{N}(i)\cup\{i\}} \boldsymbol{W}_{ij} \boldsymbol{d}_i^{(t-1)} + \alpha \nabla F_i(\boldsymbol{x}_i^{(t)}), \quad (9)$$

which leads to a worse iteration/communication complexity of $\mathcal{O}(1/\epsilon^2)$.

4.2 Decentralized Stochastic Gradient Tracking Frank-Wolfe

The DeGTFW algorithm requires to evaluate full local gradients at each iteration. However, for some applications, the evaluation of full gradients is prohibitive. One notable example is the multilinear extension of a submodular set function. For general multilinear extensions, it takes exponential time to evaluate the full local gradient $\nabla F_i(\boldsymbol{x})$. Nevertheless, one can evaluate a cheap unbiased estimate $\tilde{\nabla} F_i(\boldsymbol{x})$ in time $\mathcal{O}(d)$ (see, e.g., (Mokhtari et al., 2018b, Section 9.7)). For this situation, we propose a stochastic variant of Algorithm 1 named Decentralized Stochastic Gradient Tracking Frank-Wolfe (DeSGTFW), which is summarized in Algorithm 2.

Algorithm 2: Decentralized Stochastic Gradient Tracking Frank-Wolfe for node i

Input : matrix $\boldsymbol{W} \in \mathbb{R}_+^{n\times n}$, constraint $\mathcal{C} \subset \mathbb{R}_+^n$, number of itertions T

Initialize $\boldsymbol{x}_i^{(1)} = 0$, $\boldsymbol{g}_i^{(1)} = \tilde{\nabla}_i^{(1)} = \tilde{\nabla} F_i(\boldsymbol{x}_i^{(1)})$

for $t = 1, 2, \ldots, T$ **do**

Obtain $\boldsymbol{g}_j^{(t)}$ and $\boldsymbol{x}_j^{(t)}$ from neighbors $j \in \mathcal{N}(i)$;

$\boldsymbol{d}_i^{(t)} = \sum_{j\in\mathcal{N}(i)\cup\{i\}} \boldsymbol{W}_{ij} \cdot \boldsymbol{g}_j^{(t)}$;

$\boldsymbol{v}_i^{(t)} = \operatorname{argmax}_{\boldsymbol{v}\in\mathcal{C}} \langle \boldsymbol{v}, \boldsymbol{d}_i^{(t)} \rangle$;

$\boldsymbol{x}_i^{(t+1)} = \sum_{j\in\mathcal{N}(i)\cup\{i\}} \boldsymbol{W}_{ij} \cdot \boldsymbol{x}_j^{(t)} + \frac{1}{T}\boldsymbol{v}_i^{(t)}$;

Compute $\tilde{\nabla}_i^{(t+1)}$, the average of $(t+1)^2$ i.i.d. samples of $\tilde{\nabla} F_i(\boldsymbol{x}_i^{(t+1)})$;

$\boldsymbol{g}_i^{(t+1)} = \boldsymbol{d}_i^{(t)} + \tilde{\nabla}_i^{(t+1)} - \tilde{\nabla}_i^{(t)}$

end

Return: $\boldsymbol{x}_i^{(T+1)}$

The DeSGTFW method requires to evaluate a mini-batch (of size t^2) stochastic local gradient to construct an unbiased estimator at each node. Such increasing mini-batch size technique has been studied by Hazan and Luo (2016) and Reddi et al. (2016). In comparison, the stochastic algorithm DeSCG proposed by Mokhtari et al. (2018b) adopts another local gradient estimator: $\tilde{\nabla}_i^{(t)} = (1-\phi)\tilde{\nabla}_i^{(t-1)} + \phi\tilde{\nabla} F_i(\boldsymbol{x}_i^{(t)})$, where $\phi \in (0, 1)$ is a parameter. Such estimator requires only a fixed mini-batch of size 1.

5 Convergence Analysis

In this section, we give theoretical analyses for the proposed methods DeGTFW and DeSGTFW. We first present some lemmas that hold for both methods and then analyze their communication and gradient evaluation complexities, respectively. All the proofs in this section are deferred to **Appendix** due to the limit of space. We note that the proofs of Lemma 1, 2, and 3 are borrowed from (Mokhtari et al., 2018b) and we state them here for completeness.

To begin with, we make the following mild assumptions on the constraint set $\mathcal{C}$ and local objective functions F_i's, respectively.

Assumption 2. *The compact convex set $\mathcal{C} \subset \mathbb{R}_+^d$ has a diameter $D = \sup_{\boldsymbol{x},\boldsymbol{y}\in\mathcal{C}} \|\boldsymbol{x} - \boldsymbol{y}\|$ and a radius $R = \sup_{\boldsymbol{x}\in\mathcal{C}} \|\boldsymbol{x}\|$.*

Assumption 3. *Each of the local gradient ∇F_i at the origin $\mathbf{0}_d$ is bounded by a constant G, i.e., $\|\nabla F_i(\mathbf{0}_d)\| \le G$.*

The above assumption combined with (5) implies that $\|\nabla F_i(\boldsymbol{x})\| \le G$ for any $\boldsymbol{x} \in \mathcal{X}$ and thus F_i is G-Lipschitz over $\mathcal{X}$.

Assumption 4. *Each of the local objective functions $F_i(\cdot)$ is L-smooth over $\mathcal{X}$, i.e., $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}$*

$$\|\nabla F_i(\boldsymbol{x}) - \nabla F_i(\boldsymbol{y})\| \leq L\|\boldsymbol{x} - \boldsymbol{y}\|. \tag{10}$$

The following lemma shows that the output $\boldsymbol{x}_i^{(T+1)}$ of the proposed methods always lie in the feasible set.

Lemma 1. *The local variables $\boldsymbol{x}_i^{(t)}$ generated by Algorithm 1 and Algorithm 2 satisfy $\boldsymbol{x}_i^{(t)} \in \mathcal{X}$ and $\|\boldsymbol{x}_i^{(t)}\| \leq R$ for all $i \in [n]$, $t \in [T+1]$. Moreover, the output variable $\boldsymbol{x}_i^{(T+1)}$ lies in the feasible set $\mathcal{C}$.*

The next lemma provides an upper bound of the distance between $\boldsymbol{x}_i^{(t)}$ and their average $\bar{\boldsymbol{x}}^{(t)} = \frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_i^{(t)}$ and shows that the difference between any $F(\boldsymbol{x}_i^{(t)})$ and $F(\bar{\boldsymbol{x}}^{(t)})$ is upper bounded by $\mathcal{O}(1/T)$, which means that the convergence behavior of $\boldsymbol{x}_i^{(t)}$ is similar to that of $\bar{\boldsymbol{x}}^{(t)}$.

Lemma 2. *Consider the local variables $\boldsymbol{x}_i^{(t)}$ in Algorithm 1 or Algorithm 2 for any $i \in [n]$, $t \in [T+1]$. Let $\bar{\boldsymbol{x}}^{(t)}$ be their average, i.e., $\bar{\boldsymbol{x}}^{(t)} = \frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_i^{(t)}$, then*

$$\sqrt{\sum_{i=1}^n \|\boldsymbol{x}_i^{(t)} - \bar{\boldsymbol{x}}^{(t)}\|^2} \leq \frac{\sqrt{n}R}{T(1-\beta)}. \tag{11}$$

Furthermore, for any $i \in [n]$, $t \in [T+1]$,

$$|F(\boldsymbol{x}_i^{(t)}) - F(\bar{\boldsymbol{x}}^{(t)})| \leq \frac{\sqrt{n}GR}{T(1-\beta)}. \tag{12}$$

From now on, we analyze the convergence of $\bar{\boldsymbol{x}}^{(t)}$, from which we can deduce the results for each individual $\boldsymbol{x}_i^{(t)}$. The following lemma, referred to as the *basic ascent lemma*, indicates that the objective value $F(\bar{\boldsymbol{x}}^{(t)})$ is ascending as t increases and finally converges to $(1-1/e)F(\boldsymbol{x}^*)$ as $t \to T+1$ and $T \to \infty$ if we can bound additional terms properly, where $\boldsymbol{x}^*$ denotes an optimal solution of problem (1).

Lemma 3. *Suppose that Assumptions 1-4 hold, then, for both Algorithm 1 and Algorithm 2, the following inequality holds:*

$$\begin{aligned} &F(\boldsymbol{x}^*) - F(\bar{\boldsymbol{x}}^{(t+1)}) \\ &\leq (1 - \frac{1}{T})(F(\boldsymbol{x}^*) - F(\bar{\boldsymbol{x}}^{(t)})) + \frac{LR^2}{2T^2} \\ &+ \frac{D}{T}\|\bar{\boldsymbol{d}}^{(t)} - \nabla F(\bar{\boldsymbol{x}}^{(t)})\| + \frac{D}{nT}\sum_{i=1}^n \|\boldsymbol{d}_i^{(t)} - \bar{\boldsymbol{d}}^{(t)}\|, \end{aligned} \tag{13}$$

where $\bar{\boldsymbol{d}}^{(t)}$ is the average of $\boldsymbol{d}_i^{(t)}$ for all $i \in [n]$.

In the rest of this section, we will bound the two error terms $\|\bar{\boldsymbol{d}}^{(t)} - \nabla F(\bar{\boldsymbol{x}}^{(t)})\|$ and $\sum_{i=1}^n \|\boldsymbol{d}_i^{(t)} - \bar{\boldsymbol{d}}^{(t)}\|$ and then derive convergence rates, communication complexities, and gradient evaluation complexities for DeGTFW and DeSGTFW, respectively.

5.1 Analysis of DeGTFW

For DeGTFW, the surrogates $\boldsymbol{d}_i^{(t)}$ and their average $\bar{\boldsymbol{d}}^{(t)}$ have the following properties.

Lemma 4. *For Algorithm 1, assume that Assumption 1 holds and let $\bar{\boldsymbol{d}}^{(t)} = \frac{1}{n}\sum_{i=1}^n \boldsymbol{d}_i^{(t)}$ and $\bar{\boldsymbol{g}}^{(t)} = \frac{1}{n}\sum_{i=1}^n \boldsymbol{g}_i^{(t)}$, then for any $t \in \{1, \ldots, T\}$,*

(a) $\bar{\boldsymbol{d}}^{(t)} = \bar{\boldsymbol{g}}^{(t)} = \frac{1}{n}\sum_{i=1}^n \nabla F_i(\boldsymbol{x}_i^{(t)})$;

(b) $\sum_{i=1}^n \|\boldsymbol{d}_i^{(t)} - \bar{\boldsymbol{d}}^{(t)}\|^2 \leq \beta^2 \sum_{i=1}^n \|\boldsymbol{g}_i^{(t)} - \bar{\boldsymbol{g}}^{(t)}\|^2$.

With Lemma 4 at hand, one can easily bound the error term $\|\bar{\boldsymbol{d}}^{(t)} - \nabla F(\bar{\boldsymbol{x}}^{(t)})\|$ using Lemma 2. For the other error term $\sum_{i=1}^n \|\boldsymbol{d}_i^{(t)} - \bar{\boldsymbol{d}}^{(t)}\|$, it can be bounded from above as in the following lemma.

Lemma 5. *Consider the surrogates $\boldsymbol{d}_i^{(t)}$, $i \in [n]$ in Algorithm 1, we have for all $t \in \{1, \ldots, T\}$,*

$$\sum_{i=1}^n \|\boldsymbol{d}_i^{(t)} - \bar{\boldsymbol{d}}^{(t)}\| \leq \beta^t nG + \frac{3n\beta LR}{(1-\beta)^2 T}. \tag{14}$$

Now we are ready to present the first main theorem which shows that Algorithm 1 achieves an ϵ-accuracy tight approximation ratio.

Theorem 1. *Consider problem (1) and the DeGTFW algorithm. Suppose that Assumptions 1-4 hold, then,*

$$\begin{aligned} F(\bar{\boldsymbol{x}}^{(T+1)}) \geq (1 - \frac{1}{e})F(\boldsymbol{x}^*) - \frac{1}{T}\Big(&\frac{LRD + \beta GD}{1-\beta} \\ &+ \frac{3\beta LRD}{(1-\beta)^2} + \frac{LR^2}{2}\Big). \end{aligned} \tag{15}$$

Furthermore, all $\boldsymbol{x}_i^{(T+1)}$ for $i \in [n]$ and $\bar{\boldsymbol{x}}^{(T+1)}$ attain objective values larger than $(1-1/e)F(\boldsymbol{x}^*) - \epsilon$ after at most $\mathcal{O}(\frac{1}{\epsilon})$ communication rounds and $\mathcal{O}(\frac{1}{\epsilon})$ full local gradient evaluations.

5.2 Analysis of DeSGTFW

In this subsection, we analyze the convergence rate of DeSGTFW. First, we make the following assumption on the variance of the stochastic gradient estimator, which is common in stochastic settings.

Assumption 5. *For any $\boldsymbol{x} \in \mathcal{C}$, the variance of the stochastic, unbiased gradient estimator $\tilde{\nabla} F_i(\boldsymbol{x})$ is upper bounded by a constant, i.e.,*

$$\mathbb{E}[\|\tilde{\nabla} F_i(\boldsymbol{x}) - \nabla F_i(\boldsymbol{x})\|^2] \leq \sigma^2, \ \forall \boldsymbol{x} \in \mathcal{C}, \ i \in [n]. \tag{16}$$

Similar to the analysis of Algorithm 1, we bound $\|\bar{\boldsymbol{d}}^{(t)} - \nabla F(\bar{\boldsymbol{x}}^{(t)})\|$ and $\sum_{i=1}^n \|\boldsymbol{d}_i^{(t)} - \bar{\boldsymbol{d}}^{(t)}\|$ for DeSGTFW in the following two lemmas, respectively.

Lemma 6. *For Algorithm 2 , suppose that Assumption 1-5 hold, the error term* $\|\bar{\boldsymbol{d}}^{(t)} - \nabla F(\bar{\boldsymbol{x}}^{(t)})\|$, $t \in \{1, \dots, T\}$, *satisfies*

$$\mathbb{E}\big[\|\bar{\boldsymbol{d}}^{(t)} - \nabla F(\bar{\boldsymbol{x}}^{(t)})\|\big] \leq \frac{LR}{T(1-\beta)} + \frac{\sigma}{t}. \tag{17}$$

Lemma 7. *Define* $\tilde{M} := t_0 \cdot \max\{\beta(\sqrt{\sigma^2 + G^2} + \frac{2(\sigma+LR)}{1-\beta}), 2\sigma + \frac{3LR}{1-\beta}\}$, *where* t_0 *is the smallest integer that satisfies* $\beta \leq \frac{1}{(1+1/t_0)^2}$. *If the conditions in Lemma 6 hold, then for all* $t \in \{1, \dots, T\}$,

$$\sum_{i=1}^{n} \mathbb{E}\big[\|\boldsymbol{d}_i^{(t)} - \bar{\boldsymbol{d}}^{(t)}\|\big] \leq \frac{n\tilde{M}}{t}. \tag{18}$$

We conclude in Theorem 2 that DeSGTFW achieves an ϵ-accuracy tight approximation ratio.

Theorem 2. *Consider problem (1) and the proposed DeSGTFW algorithm. Suppose that Assumptions 1-5 hold, then*

$$\mathbb{E}[F(\bar{\boldsymbol{x}}^{(T+1)})] \geq (1 - \frac{1}{e})F(\boldsymbol{x}^*) - \frac{\log T + 1}{T}(\sigma + \tilde{M})D - \frac{1}{T}\Big(\frac{LRD}{1-\beta} + \frac{LR^2}{2}\Big). \tag{19}$$

where $\tilde{M}$ *is defined in Lemma 7. Furthermore, all* $\boldsymbol{x}_i^{(T+1)}$ *for* $i \in [n]$ *and* $\bar{\boldsymbol{x}}^{(T+1)}$ *attain objective values larger than* $(1 - 1/e)F(\boldsymbol{x}^*) - \epsilon$ *in expectation after at most* $\tilde{\mathcal{O}}(\frac{1}{\epsilon})$ *communication rounds and* $\tilde{\mathcal{O}}(\frac{1}{\epsilon^3})$ *stochastic local gradient evaluations.*

6 Experiment

In this section, we conduct numerical experiments to demonstrate the advantages of DeGTFW and DeSGTFW over existing decentralized methods DeCG and DeSCG. We compare the average objective values $\frac{1}{n}\sum_{i=1}^{n} F(\boldsymbol{x}_i^{(T+1)})$ obtained by different methods versus the amount of communication or number of full/stochastic local gradient evaluations. In particular, we repeat the stochastic methods for multiple trails and compare their results in average. We focus on two continuous DR-submodular maximization problems: a non-convex/non-concave quadratic programming problem and the multilinear extension of a submodular set function maximization problem. For all decentralized methods in our experiments, the network graph $\mathcal{G}$ is an Erdős-Rényi random graph containing $n = 50$ nodes with average degree 10. The weight matrix is computed as

$$\boldsymbol{W}_{ij} = \begin{cases} 1/(1 + \max(\deg(i), \deg(j))), & \text{if } j \in \mathcal{N}(i), \\ 1 - \sum_{\ell \in \mathcal{N}(i)} \boldsymbol{W}_{i\ell}, & \text{if } j = i, \\ 0, & \text{otherwise} \end{cases}$$

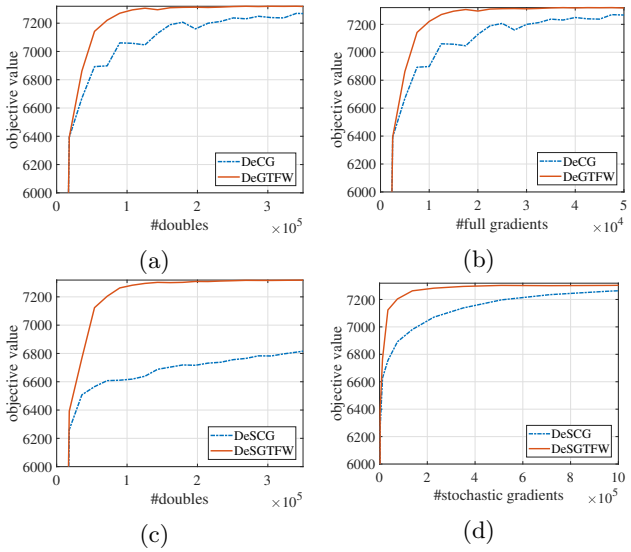

Figure 1: Comparison of DeGTFW/DeSGTFW v.s. DeCG/DeSCG on the NQP problem. The left column compares the the objective value versus the amount of communication (the total number of DOUBLEs received by all nodes until iteration T). The right column compares the objective value versus the total number of full/stochastic local gradient evaluations.

where $\deg(i)$ denotes the degree of node i. One can check that $\boldsymbol{W}$ satisfies Assumption 1. The β value of such matrix roughly lies in the range $[0.43, 0.87]$ with a mean value around 0.56.

6.1 Non-convex/non-concave Quadratic Programming

In the first experiment, we consider a synthetic NQP problem of the form

$$\max_{\boldsymbol{x} \in \mathcal{C}} F(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n}(\frac{1}{2}\boldsymbol{x}^\top \boldsymbol{H}_i \boldsymbol{x} + \boldsymbol{h}_i^\top \boldsymbol{x}), \tag{20}$$

where $\boldsymbol{h}_i \in \mathbb{R}^d$, $\boldsymbol{H}_i \in \mathbb{R}^{d \times d}$, and each pair of $(\boldsymbol{H}_i, \boldsymbol{h}_i)$ is stored on node i. Following the convention of (Chen et al., 2018), we generate $(\boldsymbol{H}_i, \boldsymbol{h}_i)$ and construct the constraint set as follows. Each entry of $\boldsymbol{H}_i$ is uniformly sampled from $[-100, 0]$ so that F_i is DR-submodular. The constraint set $\mathcal{C} = \{\boldsymbol{0}_d \leq \boldsymbol{x} \leq \boldsymbol{1}_d, \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{1}_d\}$, where each entry in $\boldsymbol{A} \in \mathbb{R}^{m \times d}$ is uniformly sampled from $[0, 1]$. To ensure that F_i is monotone over $\mathcal{C}$, we set $\boldsymbol{h}_i = -\boldsymbol{H}_i^\top \cdot \boldsymbol{1}_d$. We set $d = 20$ and $m = 2$.

We consider different settings for the deterministic methods and stochastic methods, respectively. In the comparison of the deterministic methods DeGTFW and DeCG, we assume that each node i is able to evaluate the full local gradient $\nabla F_i(\boldsymbol{x}) = \boldsymbol{H}_i\boldsymbol{x} + \boldsymbol{h}_i$. As for the stochastic methods, each node i is only provided

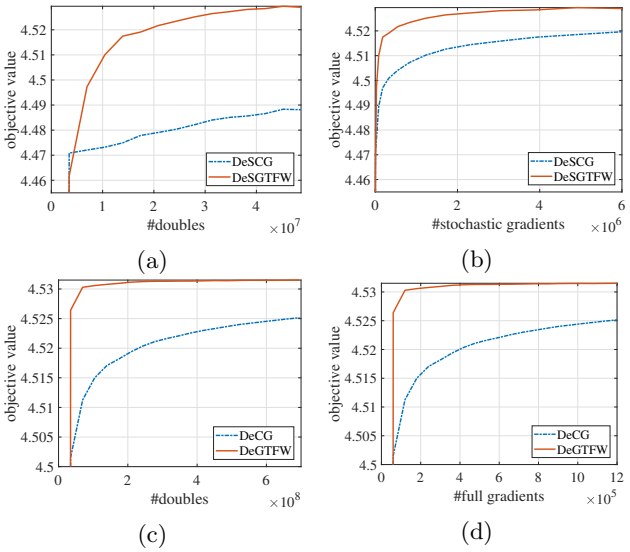

Figure 2: Results for the movie recommendation problem. The first row compares the stochastic methods DeSGTFW and DeSCG. The second row compares the deterministic methods DeGTFW and DeCG.

with an unbiased estimator $\tilde{\nabla} F_i(\boldsymbol{x}) = \boldsymbol{H}_i \boldsymbol{x} + \boldsymbol{h}_i + \boldsymbol{\xi}$, where $\boldsymbol{\xi} \sim N(\boldsymbol{0}_d, 10 \cdot \boldsymbol{I}_d)$, the Gaussian distribution with mean $\boldsymbol{0}_d$ and covariance $10 \cdot \boldsymbol{I}_d$.

The results are shown in Figure 1. It can be seen from Figure 1a and 1b that DeGTFW outperforms DeCG in terms of both communication and gradient evaluation complexities to reach the same objective value. Similarly, Figure 1c and 1d show that DeSGTFW outperforms DeSCG in terms of both communication and stochastic gradient evaluation complexities.

6.2 Movie Recommendation

The second experiment we consider here is a real-world movie recommendation application (Stan et al., 2017; Hassani et al., 2017), which aims to recommend a set of $k = 10$ movies to all users. The data set we use is the "MovieLens1M" data set[4], which contains 1 million entries of rating ranging from 1 to 5 from 6,000 users on $d = 4,000$ movies. We use $r(u, m)$ to denote user u's rating for movie $m \in [d]$ and set $r(u, m) = 0$ if movie m is not rated by user u. The whole data set is split into $n = 50$ batches $\{B_1, \ldots, B_n\}$, where each batch B_i contains the ratings of exactly $b = 120$ users. Then we distribute $\{B_1, \ldots, B_n\}$ onto n computing nodes in a network. The goal is to maximize the multilinear extension $F(\boldsymbol{x}) = \frac{1}{n} \sum_{i=1}^{n} F_i(\boldsymbol{x})$ of a submodular set function. Specifically, F_i is defined as

$$F_i(\boldsymbol{x}) = \sum_{S \subset [d]} f_i(S) \prod_{j \in S} [\boldsymbol{x}]_j \prod_{\ell \notin S} (1 - [\boldsymbol{x}]_\ell), \ \boldsymbol{x} \in \mathcal{C}. \quad (21)$$

Here, $[\boldsymbol{x}]_\ell$ denotes the ℓ-th component of $\boldsymbol{x}$ (with a slight abuse of notation), $\mathcal{C} = \{\boldsymbol{x} \in [0,1]^d, \sum_{i=1}^{d} [\boldsymbol{x}]_i = k\}$, and $f_i(S)$ is the facility location function, i.e.,

$$f_i(S) = \frac{1}{|B_i|} \sum_{u \in B_i} \max_{m \in S} \ r(u, m), \quad (22)$$

subject to $S \subset [d]$ and $|S| = k$. It can be verified that F_i is a monotone DR-submodular function and the feasible set $\mathcal{C}$ is a compact convex set. We can observe from (21) that both the computations of F_i and ∇F_i require 2^d evaluations of the discrete function f_i, which is prohibitive when d is large. Nevertheless, one can construct an unbiased gradient estimator $\tilde{\nabla} F_i(\boldsymbol{x})$ as follows. First, we sample a random set $Q \subset [d]$, where each element $\ell \in [d]$ is included in Q independently with probability $[\boldsymbol{x}]_\ell$. Then the j-th component of $\tilde{\nabla} F_i$ is computed as $[\tilde{\nabla} F_i]_j = f_i(Q \cup \{j\}) - f_i(Q \backslash \{j\})$ (see, e.g., (Calinescu et al., 2011, Section 2)).

In Figure 2a and 2b, we compare the stochastic method DeSGTFW with the baseline DeSCG (Mokhtari et al., 2018b). It can be observed that given the same amount of communication or the same number of stochastic gradient evaluations, DeSGTFW beats DeSCG as it attains a larger objective value.

Actually, for this movie recommendation task, there is another formulation of (21) such that ∇F_i can be computed in polynomial time (Iyer et al., 2014). Thus, we also compare the performance of DeGTFW and DeCG for this task. The results are shown in Figure 2c and 2d. We can see that DeGTFW converges faster than DeCG in terms of both the amount of communication and the number of full gradient evaluations.

7 Conclusion

In this paper, we propose two decentralized consensus methods, DeGTFW and DeSGTFW, for solving large-scale monotone, continuous DR-submodular maximization problems subject to any compact convex set. We theoretically analyze the proposed methods, which shows that our methods are superior to existing methods in terms of communication complexity. The numerical results also validate the advantages of our methods over existing ones.

Acknowledgements

This work is supported by Zhejiang Provincial Natural Science Foundation of China under Grant No.

[4]The data set can be downloaded from https://grouplens.org/datasets/movielens/.

LZ18F020002, and National Natural Science Foundation of China (Grant No: 61472347, 61672376, 61751209).

References

Ageev, A. A. and Sviridenko, M. I. (2004). Pipage rounding: A new method of constructing algorithms with proven performance guarantee. *Journal of Combinatorial Optimization*, 8(3):307–328.

Azar, Y. and Gamzu, I. (2012). Efficient submodular function maximization under linear packing constraints. In *International Colloquium on Automata, Languages, and Programming*, pages 38–50. Springer.

Barbosa, R., Ene, A., Nguyen, H., and Ward, J. (2015). The power of randomization: Distributed submodular maximization on massive datasets. In *International Conference on Machine Learning*, pages 1236–1244.

Bian, A. A., Mirzasoleiman, B., Buhmann, J. M., and Krause, A. (2017). Guaranteed non-convex optimization: Submodular maximization over continuous domains. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 111–120.

Calinescu, G., Chekuri, C., Pál, M., and Vondrák, J. (2011). Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*, 40(6):1740–1766.

Chen, L., Hassani, H., and Karbasi, A. (2018). Online continuous submodular maximization. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, pages 1896–1905.

Defazio, A., Bach, F., and Lacoste-Julien, S. (2014). Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in neural information processing systems*, pages 1646–1654.

Di Lorenzo, P. and Scutari, G. (2016). Next: In-network nonconvex optimization. *IEEE Transactions on Signal and Information Processing over Networks*, 2(2):120–136.

Eghbali, R. and Fazel, M. (2016). Designing smoothing functions for improved worst-case competitive ratio in online optimization. In *Advances in Neural Information Processing Systems*, pages 3287–3295.

El-Arini, K., Veda, G., Shahaf, D., and Guestrin, C. (2009). Turning down the noise in the blogosphere. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 289–298. ACM.

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

Gharesifard, B. and Smith, S. L. (2016). On distributed submodular maximization with limited information. In *American Control Conference (ACC), 2016*, pages 1048–1053. IEEE.

Golovin, D., Faulkner, M., and Krause, A. (2010). Online distributed sensor selection. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, pages 220–231. ACM.

Golovin, D. and Krause, A. (2011). Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *Journal of Artificial Intelligence Research*, 42:427–486.

Golovin, D., Krause, A., and Streeter, M. (2014). Online submodular maximization under a matroid constraint with application to learning assignments. *arXiv preprint arXiv:1407.1082*.

Hassani, H., Soltanolkotabi, M., and Karbasi, A. (2017). Gradient methods for submodular maximization. In *Advances in Neural Information Processing Systems*, pages 5841–5851.

Hazan, E. and Luo, H. (2016). Variance-reduced and projection-free stochastic optimization. In *ICML*, volume 16, pages 1263–1271.

Iyer, R., Jegelka, S., and Bilmes, J. (2014). Monotone closure of relaxed constraints in submodular optimization: Connections between minimization and maximization. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, UAI'14.

Krause, A. and Golovin, D. (2014). *Submodular Function Maximization*, page 71–104.

Krause, A., Rajagopal, R., Gupta, A., and Guestrin, C. (2009). Simultaneous placement and scheduling of sensors. In *Proceedings of the 2009 International Conference on Information Processing in Sensor Networks*, pages 181–192. IEEE Computer Society.

Kumar, R., Moseley, B., Vassilvitskii, S., and Vattani, A. (2015). Fast greedy algorithms in mapreduce and streaming. *ACM Transactions on Parallel Computing (TOPC)*, 2(3):14.

Lin, H. and Bilmes, J. (2010). Multi-document summarization via budgeted maximization of submodular functions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 912–920. Association for Computational Linguistics.

Mirzasoleiman, B., Karbasi, A., Sarkar, R., and Krause, A. (2013). Distributed submodular maximization: Identifying representative elements in massive data. In *Advances in Neural Information Processing Systems*, pages 2049–2057.

Mokhtari, A., Hassani, H., and Karbasi, A. (2018a). Conditional gradient method for stochastic submodular maximization: Closing the gap. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, pages 1886–1895.

Mokhtari, A., Hassani, H., and Karbasi, A. (2018b). Decentralized submodular maximization: Bridging discrete and continuous settings. In *Proceedings of the 35th International Conference on Machine Learning*, pages 3616–3625.

Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L. (1978). An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14(1):265–294.

Pu, S. and Nedić, A. (2018). A distributed stochastic gradient tracking method. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 963–968. IEEE.

Qu, G. and Li, N. (2017). Accelerated distributed nesterov gradient descent. *arXiv preprint arXiv:1705.07176*.

Reddi, S. J., Sra, S., Póczos, B., and Smola, A. (2016). Stochastic frank-wolfe methods for nonconvex optimization. In *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1244–1251. IEEE.

Singh, A., Krause, A., Guestrin, C., Kaiser, W. J., and Batalin, M. A. (2007). Efficient planning of informative paths for multiple robots. In *IJCAI*, volume 7, pages 2204–2211.

Staib, M. and Jegelka, S. (2017). Robust budget allocation via continuous submodular functions. *arXiv preprint arXiv:1702.08791*.

Stan, S., Zadimoghaddam, M., Krause, A., and Karbasi, A. (2017). Probabilistic submodular maximization in sub-linear time. In *International Conference on Machine Learning*, pages 3241–3250.

Sviridenko, M. (2004). A note on maximizing a submodular set function subject to a knapsack constraint. *Operations Research Letters*, 32(1):41–43.

Vondrák, J. (2008). Optimal approximation for the submodular welfare problem in the value oracle model. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 67–74. ACM.

Wai, H.-T., Lafond, J., Scaglione, A., and Moulines, E. (2017). Decentralized frank–wolfe algorithm for convex and nonconvex problems. *IEEE Transactions on Automatic Control*, 62(11):5522–5537.

Wolsey, L. A. (1982). An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 2(4):385–393.

Yuan, K., Ling, Q., and Yin, W. (2016). On the convergence of decentralized gradient descent. *SIAM Journal on Optimization*, 26(3):1835–1854.

Yue, Y. and Guestrin, C. (2011). Linear submodular bandits and their application to diversified retrieval. In *Advances in Neural Information Processing Systems*, pages 2483–2491.