**Noname manuscript No.**
(will be inserted by the editor)

# Communication-Efficient Algorithms for Decentralized and Stochastic Optimization

**Guanghui Lan · Soomin Lee · Yi Zhou**

the date of receipt and acceptance should be inserted later

**Abstract** We present a new class of decentralized first-order methods for nonsmooth and stochastic optimization problems defined over multiagent networks. Considering that communication is a major bottleneck in decentralized optimization, our main goal in this paper is to develop algorithmic frameworks which can significantly reduce the number of inter-node communications. We first propose a decentralized primal-dual method which can find an $\epsilon$-solution both in terms of functional optimality gap and feasibility residual in $\mathcal{O}(1/\epsilon)$ inter-node communication rounds when the objective functions are convex and the local primal subproblems are solved exactly. Our major contribution is to present a new class of decentralized primal-dual type algorithms, namely the decentralized communication sliding (DCS) methods, which can skip the inter-node communications while agents solve the primal subproblems iteratively through linearizations of their local objective functions. By employing DCS, agents can still find an $\epsilon$-solution in $\mathcal{O}(1/\epsilon)$ (resp., $\mathcal{O}(1/\sqrt{\epsilon})$) communication rounds for general convex functions (resp., strongly convex functions), while maintaining the $\mathcal{O}(1/\epsilon^2)$ (resp., $\mathcal{O}(1/\epsilon)$) bound on the total number of intra-node subgradient evaluations. We also present a stochastic counterpart for these algorithms, denoted by SDCS, for solving stochastic optimization problems whose objective function cannot be evaluated exactly. In comparison with existing results for decentralized nonsmooth and stochastic optimization, we can reduce the total number of inter-node communication rounds by orders of magnitude while still maintaining the optimal complexity bounds on intra-node stochastic subgradient evaluations. The bounds on the (stochastic) subgradient evaluations are actually comparable to those required for centralized nonsmooth and stochastic optimization under certain conditions on the target accuracy.

**Keywords:** decentralized optimization, decentralized machine learning, communication efficient, stochastic programming, nonsmooth functions, primal-dual method, complexity

**AMS 2000 subject classification:** 90C25, 90C06, 90C22, 49M37, 93A14, 90C15

## 1 Introduction

Decentralized optimization problems defined over complex multiagent networks are ubiquitous in signal processing, machine learning, control, and other areas in science and engineering (see e.g. [47,21,50,15]). In this paper, we consider the following decentralized optimization problem which is cooperatively solved by the network of $m$ agents:

$$\begin{aligned} &\min_x \ f(x) := \textstyle\sum_{i=1}^m f_i(x) \\ &\text{s.t.} \ \ x \in X, \quad X := \cap_{i=1}^m X_i, \end{aligned} \tag{1.1}$$

This work was funded by National Science Foundation grants 1637473 and 1637474, and Office of Naval Research grant N00014-16-1-2802

Department of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, 30332. (E-mail: george.lan@isye.gatech.edu, soomin.lee@isye.gatech.edu, yizhou@gatech.edu)

Address(es) of author(s) should be given

where $f_i : X_i \to \mathbb{R}$ is a convex and possibly nonsmooth objective function of agent $i$ satisfying

$$\tfrac{\mu}{2}\|x-y\|^2 \le f_i(x) - f_i(y) - \langle f_i'(y), x-y\rangle \le M\|x-y\|, \quad \forall x, y \in X_i, \tag{1.2}$$

for some $M, \mu \ge 0$ and $f_i'(y) \in \partial f_i(y)$, where $\partial f_i(y)$ denotes the subdifferential of $f_i$ at $y$, and $X_i \subseteq \mathbb{R}^d$ is a closed convex constraint set of agent $i$. Note that $f_i$ and $X_i$ are private and only known to agent $i$. Throughout the paper, we assume the feasible set $X$ is nonempty.

In this paper, we also consider the situation where one can only have access to noisy first-order information (function values and subgradients) of the functions $f_i$, $i = 1, \ldots, m$ (see [41,23]). This happens, for example, when the function $f_i$'s are given in the form of expectation, i.e.,

$$f_i(x) := \mathbb{E}_{\xi_i}[F_i(x; \xi_i)], \tag{1.3}$$

where the random variable $\xi_i$ models a source of uncertainty and the distribution $\mathbb{P}(\xi_i)$ is not known in advance. As a special case of (1.3), $f_i$ may be given as the summation of many components, i.e.,

$$f_i(x) := \textstyle\sum_{j=1}^{l} f_i^j(x), \tag{1.4}$$

where $l \ge 1$ is some large number. Stochastic optimization problem of this type has great potential of applications in data analysis, especially in machine learning. In particular, problem (1.3) corresponds to the minimization of generalized risk and is particularly useful for dealing with online (streaming) data distributed over a network, while problem (1.4) aims at the collaborative minimization of empirical risk. Currently the dominant approach is to collect all agents' private data on a server (or cluster) and to apply centralized machine learning techniques. However, this centralization scheme would require agents to submit their private data to the service provider without much control on how the data will be used, in addition to incurring high setup cost related to the transmission of data to the service provider. Decentralized optimization provides a viable approach to deal with these data privacy related issues.

In these decentralized and stochastic optimization problems, each network agent $i$ is associated with the local objective function $f_i(x)$ and all agents intend to cooperatively minimize the system objective $f(x)$ as the sum of all local objective $f_i$'s in the absence of full knowledge about the global problem and network structure. A necessary feature in decentralized optimization is, therefore, that the agents must communicate with their neighboring agents to propagate the distributed information to every location in the network.

One of the most well-studied techniques in decentralized optimization are the subgradient based methods (see e.g., [39,35,57,37,14,27,52]), where at each step a local subgradient is taken at each node, followed by the communication with neighboring agents. Although the subgradient computation at each step can be inexpensive, these methods usually require lots of iterations until convergence. Considering that one iteration in decentralized optimization is equivalent to one communication round among agents, this can incur a significant latency. CPUs in these days can read and write the memory at over 10 GB per second whereas communication over TCP/IP is about 10 MB per second. Therefore, the gap between intra-node computation and inter-node communication is about 3 orders of magnitude. The communication start-up cost itself is also not negligible as it usually takes a few milliseconds.

Another well-known type of decentralized algorithm relies on dual methods (see e.g., [4,62,10]), where at each step for a fixed dual variable, the primal variables are solved to minimize some local Lagrangian related function, then the dual variables associated with the consistency constraints are updated accordingly. Although these dual type methods usually require fewer numbers of iterations (hence, fewer communication rounds) than the subgradient methods until convergence, one crucial problem of these methods is that the local subproblem associated with each agent cannot be solved efficiently in many cases.

The main goal of this paper is, therefore, to develop dual based decentralized algorithms for solving (1.1) that is communication efficient and has local subproblems easily solved by each agent through the utilization of (noisy) first-order information of $f_i$. More specifically, we will provide a theoretical understanding on how many numbers of inter-node communications and intra-node (stochastic) subgradient evaluations of $f_i$ are required in order to find a certain approximate solution of (1.1).

### 1.1 Notation and Terminologies

Let $\mathbb{R}$ denote the set of real numbers. All vectors are viewed as column vectors, and for a vector $x \in \mathbb{R}^d$, we use $x^\top$ to denote its transpose. For a stacked vector of $x_i$'s, we often use $(x_1, \ldots, x_m)$ to represent the column vector $[x_1^\top, \ldots, x_m^\top]^\top$. We denote by $\mathbf{0}$ and $\mathbf{1}$ the vector of all zeros and ones whose dimensions vary from the context. The cardinality of a set $S$ is denoted by $|S|$. We use $I_d$ to denote the identity matrix in $\mathbb{R}^{d\times d}$. We use $A \otimes B$ for matrices $A \in \mathbb{R}^{n_1\times n_2}$ and $B \in \mathbb{R}^{m_1\times m_2}$ to denote their Kronecker product of size $\mathbb{R}^{n_1 m_1 \times n_2 m_2}$. For a matrix $A \in \mathbb{R}^{n\times m}$, we use $A_{ij}$ to denote the entry of $i$-th row and $j$-th column. For any $m \geq 1$, the set of integers $\{1, \ldots, m\}$ is denoted by $[m]$.

### 1.2 Problem Formulation

Consider a multiagent network system whose communication is governed by an undirected graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N} = [m]$ indexes the set of agents, and $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ represents the pairs of communicating agents. If there exists an edge from agent $i$ to $j$ which we denote by $(i, j)$, agent $i$ may send its information to agent $j$ and vice versa. Thus, each agent $i \in \mathcal{N}$ can directly receive (resp., send) information only from (resp., to) the agents in its neighborhood

$$N_i = \{j \in \mathcal{N} \mid (i, j) \in \mathcal{E}\} \cup \{i\}, \tag{1.5}$$

where we assume that there always exists a self-loop $(i, i)$ for all agents $i \in \mathcal{N}$. Then, the associated Laplacian $L \in \mathbb{R}^{m\times m}$ of $\mathcal{G}$ is $L := D - A$ where $D$ is the diagonal degree matrix, and $A \in \mathbb{R}^{m\times m}$ is the adjacency matrix with the property that $A_{ij} = 1$ if and only if $(i, j) \in \mathcal{E}$ and $i \neq j$, i.e.,

$$L_{ij} = \begin{cases} |N_i| - 1 & \text{if } i = j \\ -1 & \text{if } i \neq j \text{ and } (i, j) \in \mathcal{E} \\ 0 & \text{otherwise.} \end{cases} \tag{1.6}$$

We consider a reformulation of problem (1.1) which will be used in the development of our decentralized algorithms. We introduce an individual copy $x_i$ of the decision variable $x$ for each agent $i \in \mathcal{N}$ and impose the constraint $x_i = x_j$ for all pairs $(i, j) \in \mathcal{E}$. The transformed problem can be written compactly by using the Laplacian matrix $L$:

$$\begin{aligned} &\min_{\mathbf{x}} \ F(\mathbf{x}) := \textstyle\sum_{i=1}^m f_i(x_i) \\ &\text{s.t.} \ \ \mathbf{L}\mathbf{x} = \mathbf{0}, \quad x_i \in X_i, \ \text{for all } i = 1, \ldots, m, \end{aligned} \tag{1.7}$$

where $\mathbf{x} = (x_1, \ldots, x_m) \in X_1 \times \ldots \times X_m$, $F : X_1 \times \ldots \times X_m \to \mathbb{R}$, and $\mathbf{L} = L \otimes I_d \in \mathbb{R}^{md\times md}$. The constraint $\mathbf{L}\mathbf{x} = \mathbf{0}$ is a compact way of writing $x_i = x_j$ for all agents $i$ and $j$ which are connected by an edge. By construction, $\mathbf{L}$ is symmetric positive semidefinite and its null space coincides with the "agreement" subspace, i.e., $\mathbf{L}\mathbf{1} = \mathbf{0}$ and $\mathbf{1}^\top \mathbf{L} = \mathbf{0}$. To ensure each node gets information from every other node, we need the following assumption.

**Assumption 1** *The graph $\mathcal{G}$ is connected.*

Under Assumption 1, problem (1.1) and (1.7) are equivalent. We let Assumption 1 be a blanket assumption for the rest of the paper.

We next consider a reformulation of the problem (1.7) as a saddle point problem. By the method of Lagrange multipliers, problem (1.7) is equivalent to the following saddle point problem:

$$\min_{\mathbf{x}\in X^m} \left[ F(\mathbf{x}) + \max_{\mathbf{y}\in\mathbb{R}^{md}} \langle \mathbf{L}\mathbf{x}, \mathbf{y}\rangle \right], \tag{1.8}$$

where $X^m := X_1 \times \ldots \times X_m$ and $\mathbf{y} = (y_1, \ldots, y_m) \in \mathbb{R}^{md}$ are the Lagrange multipliers associated with the constraints $\mathbf{L}\mathbf{x} = \mathbf{0}$. We assume that there exists an optimal solution $\mathbf{x}^* \in X^m$ of (1.7) and that there exists $\mathbf{y}^* \in \mathbb{R}^{md}$ such that $(\mathbf{x}^*, \mathbf{y}^*)$ is a saddle point of (1.8).

### 1.3 Literature review

Decentralized optimization has been extensively studied in recent years due to the emergence of large-scale networks. The seminal work on distributed optimization [60,59] has been followed by distributed incremental (sub)gradient methods and proximal methods [36,48,2,61], and more recently the incremental aggregated gradient methods and its proximal variants [18,3,26]. All of these incremental methods are not fully decentralized in a sense that they require a special star network topology in which the existence of a central authority is necessary for operation.

To consider a more general network topology, a decentralized subgradient algorithm was first proposed in [39], and further studied in many other literature (see e.g. [14,65,35,37,56]). These algorithms are intuitive and simple but very slow due to the fact that they need to use diminishing stepsize rules. All of these methods require $\mathcal{O}(1/\epsilon^2)$ inter-node communications and intra-node gradient computations in order to obtain an $\epsilon$-optimal solution. First-order algorithms by Shi et. al. [52,53] use constant stepsize rules with backtracking and require $\mathcal{O}(1/\epsilon)$ communications when the objective function in (1.1) is a relatively simple convex function, but require both smoothness and strong convexity in order to achieve a linear convergence rate. Recently, it has been shown in [45,38] that the linear rate of convergence can be obtained for minimizing "unconstrained" smooth and strongly convex problems. These methods do not apply to general nonsmooth and stochastic optimization problems to be studied in this work.

Another well-known type of decentralized algorithm is based on dual methods including the distributed dual decomposition [55] and decentralized alternating direction method of multipliers (ADMM) [51,28,62]. The decentralized ADMM [51,28] has been shown to require $\mathcal{O}(\log 1/\epsilon)$ communications in order to obtain an $\epsilon$-optimal solution under the no constraint, strong convexity and smoothness assumptions while [62] has been shown to require $\mathcal{O}(1/\epsilon)$ communications for relatively simple convex functions $f_i$ (see also [20] for the application of mirror-prox method for solving these problems). These dual-based methods have been further studied via proximal-gradient [9,8]. However, the local Lagrangian minimization problem associated with each agent cannot be solved efficiently in many cases, especially when the problem is constrained. Second-order approximation methods [29,30] have been studied in order to handle this issue, but due to the nature of these methods differentiability of the objective function is necessary in this case.

There exist some distributed methods that just assume smoothness on the objective functions, but actually require more communication rounds than gradient computations. For example, the distributed Nesterov's accelerated gradient method [22] employs multi-consensus in the inner-loop. Although their method requires $\mathcal{O}(1/\sqrt{\epsilon})$ intra-node gradient computations, inter-node communications must increase at a rate of $\mathcal{O}(\log(k))$ as the iteration $k$ increases. Similarly, the proximal gradient method with adapt-then-combine (ATC) multi-consensus strategy and Nesterov's acceleration under the assumption of bounded and Lipschitz gradients [11] is shown to have $\mathcal{O}(1/\sqrt{\epsilon})$ intra-node gradient computations, but inter-node communications must increase at a rate of $\mathcal{O}(k)$. Due to the nature of decentralized networked systems, the time required for inter-node communications is higher by a few orders of magnitude than that for intra-node computations. Multi-consensus schemes in nested loop algorithms do not account for this feature of networked systems and hence are less desirable.

Decentralized stochastic optimization methods can be useful when the noisy gradient information of the function $f_i$, $i = 1, \ldots, m$, in (1.1) is only available or easier to compute. Stochastic first-order methods for problem (1.1) are studied in [14,49,35], all of which require $\mathcal{O}(1/\epsilon^2)$ inter-node communications and intra-node gradient computations to obtain an $\epsilon$-optimal solution. Multiagent mirror descent method for decentralized stochastic optimization [46] showed a $\mathcal{O}(1/\epsilon)$ complexity bound when the objective functions are strongly convex. An alternative form of mirror descent in the multiagent setting was proposed by [63] with an asymptotic convergence result. On a broader scale, decentralized stochastic optimization was also considered in the case of time-varying objective functions in the recent work [54,58]. All these previous works in decentralized stochastic optimization suffered from high communication costs due to the coupled scheme for stochastic subgradient evaluation and communication, i.e., each evaluation of stochastic subgradient will incur one round of communication.

### 1.4 Contribution of the paper

The main interest of this paper is to develop communication efficient decentralized algorithms for solving problem (1.7) in which $f_i$'s are convex or strongly convex, but not necessarily smooth, and the local subproblem associated with each agent is nontrivial to solve. Our contributions in this paper are listed below.

Firstly, we propose a decentralized primal-dual framework which involves only two inter-node communications per iteration. The proposed method can find an $\epsilon$-optimal solution both in terms of the primal optimality gap and feasibility residual in $\mathcal{O}(1/\epsilon)$ communication rounds when the objective functions are convex, and the local proximal projection subproblems can be solved exactly. This algorithm serves as a benchmark in terms of the communication cost for our subsequent development.

Secondly, we introduce a new decentralized primal-dual type method, called decentralized communication sliding (DCS), where the agents can skip communications while solving their local subproblems iteratively through successive linearizations of their local objective functions. We show that agents can still find an $\epsilon$-optimal solution in $\mathcal{O}(1/\epsilon)$ (resp., $\mathcal{O}(1/\sqrt{\epsilon})$) communication rounds while maintaining the $\mathcal{O}(1/\epsilon^2)$ (resp., $\mathcal{O}(1/\epsilon)$) bound on the total number of intra-node subgradient evaluations when the objective functions are general convex (resp., strongly convex). The bounds on the subgradient evaluations are actually comparable to those optimal complexity bounds required for centralized nonsmooth optimization under certain conditions on the target accuracy, and hence are not improvable in general.

Thirdly, we present a stochastic decentralized communication sliding method, denoted by SDCS, for solving stochastic optimization problems and show complexity bounds similar to those of DCS on the total number of required communication rounds and stochastic subgradient evaluations. In particular, only $\mathcal{O}(1/\epsilon)$ (resp., $\mathcal{O}(1/\sqrt{\epsilon})$) communication rounds are required while agents perform up to $\mathcal{O}(1/\epsilon^2)$ (resp., $\mathcal{O}(1/\epsilon)$) stochastic subgradient evaluations for general convex (resp., strongly convex) functions. Only requiring the access to stochastic subgradient at each iteration, SDCS is particularly efficient for solving problems with $f_i$ given in the form of (1.3) and (1.4). In the former case, SDCS requires only one realization of the random variable at each iteration and provides a communication-efficient way to deal with streaming data and decentralized machine learning. In the latter case, each iteration of SDCS requires only one randomly selected component, leading up to a factor of $\mathcal{O}(l)$ savings on the total number of subgradient computations over DCS.

To the best of our knowledge, this is the first time that these communication sliding algorithms, and the aforementioned separate complexity bounds on communication rounds and (stochastic) subgradient evaluations are presented in the literature.

### 1.5 Organization of the paper

This paper is organized as follows. In Section 2, we provide some preliminaries on distance generating functions and prox-functions, as well as the definition of gap functions, which will be used as termination criteria of our primal-dual methods. In Section 3, we present a new decentralized primal-dual method for solving problem (1.8). In Section 4, we present the communication sliding algorithms when the exact subgradients of $f_i$'s are available and establish their convergence properties for the general and strongly convex case. In Section 5, we generalize the algorithms in Section 4 for stochastic problems. The proofs of the lemmas in Section 3-5 are provided in Section 6. Finally, we provide some concluding remarks in Section 7.

## 2 Preliminaries

In this section, we provide a brief review on the prox-function, and define appropriate gap functions which will be used for the convergence analysis and termination criteria of our primal-dual algorithms.

### 2.1 Distance Generating Function and Prox-function

In this subsection, we define the concept of prox-function, which is also known as proximity control function or Bregman distance function [5]. Prox-function has played an important role in the recent development of first-order methods for convex programming as a substantial generalization of the Euclidean projection. Unlike the standard projection operator $\mathit{\Pi}_U[x] := \operatorname{argmin}_{u\in U}\|x-u\|^2$, which is inevitably tied to the Euclidean geometry, prox-function can be flexibly tailored to the geometry of a constraint set $U$.

For any convex set $U$ equipped with an arbitrary norm $\|\cdot\|_U$, we say that a function $\omega : U \to \mathbb{R}$ is a *distance generating function* with modulus $\nu > 0$ with respect to $\|\cdot\|_U$, if $\omega$ is continuously differentiable and strongly convex

with modulus $\nu$ with respect to $\|\cdot\|_U$, i.e.,

$$\langle \nabla\omega(x) - \nabla\omega(u), x - u\rangle \geq \nu\|x-u\|_U^2, \quad \forall x, u \in U. \tag{2.1}$$

The *prox-function*, or *Bregman distance function*, induced by $\omega$ is given by

$$V(x,u) \equiv V_\omega(x,u) := \omega(u) - [\omega(x) + \langle \nabla\omega(x), u - x\rangle]. \tag{2.2}$$

It then follows from the strong convexity of $\omega$ that

$$V(x,u) \geq \tfrac{\nu}{2}\|x-u\|_U^2, \quad \forall x, u \in U.$$

We now assume that the individual constraint set $X_i$ for each agent in problem (1.1) are equipped with norm $\|\cdot\|_{X_i}$, and their associated prox-functions are given by $V_i(\cdot,\cdot)$. Moreover, we assume that each $V_i(\cdot,\cdot)$ shares the same strongly convex modulus $\nu = 1$, i.e.,

$$V_i(x_i,u_i) \geq \tfrac{1}{2}\|x_i - u_i\|_{X_i}^2, \quad \forall x_i, u_i \in X_i,\ i = 1,\ldots,m. \tag{2.3}$$

We define the norm associated with the primal feasible set $X^m = X_1 \times \ldots \times X_m$ of (1.8) as follows:[1]

$$\|\mathbf{x}\|^2 \equiv \|\mathbf{x}\|_{X^m}^2 := \textstyle\sum_{i=1}^m \|x_i\|_{X_i}^2, \tag{2.4}$$

where $\mathbf{x} = (x_1,\ldots,x_m) \in X^m$ for any $x_i \in X_i$. Therefore, the corresponding prox-function $\mathbf{V}(\cdot,\cdot)$ can be defined as

$$\mathbf{V}(\mathbf{x},\mathbf{u}) := \textstyle\sum_{i=1}^m V_i(x_i,u_i),\ \forall \mathbf{x},\mathbf{u} \in X^m. \tag{2.5}$$

Note that by (2.3) and (2.4), it can be easily seen that

$$\mathbf{V}(\mathbf{x},\mathbf{u}) \geq \tfrac{1}{2}\|\mathbf{x}-\mathbf{u}\|^2,\ \forall \mathbf{x},\mathbf{u} \in X^m. \tag{2.6}$$

Throughout the paper, we endow the dual space where the multipliers $\mathbf{y}$ of (1.8) reside with the standard Euclidean norm $\|\cdot\|_2$, since the feasible region of $\mathbf{y}$ is unbounded. For simplicity, we often write $\|\mathbf{y}\|$ instead of $\|\mathbf{y}\|_2$ for a dual multiplier $\mathbf{y} \in \mathbb{R}^{md}$.

### 2.2 Gap Functions: Termination Criteria

Given a pair of feasible solutions $\mathbf{z} = (\mathbf{x},\mathbf{y})$ and $\bar{\mathbf{z}} = (\bar{\mathbf{x}},\bar{\mathbf{y}})$ of (1.8), we define the *primal-dual gap function* $Q(\mathbf{z};\bar{\mathbf{z}})$ by

$$Q(\mathbf{z};\bar{\mathbf{z}}) := F(\mathbf{x}) + \langle \mathbf{L}\mathbf{x}, \bar{\mathbf{y}}\rangle - [F(\bar{\mathbf{x}}) + \langle \mathbf{L}\bar{\mathbf{x}}, \mathbf{y}\rangle]. \tag{2.7}$$

Sometimes we also use the notations $Q(\mathbf{z};\bar{\mathbf{z}}) := Q(\mathbf{x},\mathbf{y};\bar{\mathbf{x}},\bar{\mathbf{y}})$ or $Q(\mathbf{z};\bar{\mathbf{z}}) := Q(\mathbf{x},\mathbf{y};\bar{\mathbf{z}}) = Q(\mathbf{z};\bar{\mathbf{x}},\bar{\mathbf{y}})$. One can easily see that $Q(\mathbf{z}^*;\mathbf{z}) \leq 0$ and $Q(\mathbf{z};\mathbf{z}^*) \geq 0$ for all $\mathbf{z} \in X^m \times \mathbb{R}^{md}$, where $\mathbf{z}^* = (\mathbf{x}^*,\mathbf{y}^*)$ is a saddle point of (1.8). For compact sets $X^m \subset \mathbb{R}^{md}$, $Y \subset \mathbb{R}^{md}$, the gap function

$$\sup_{\bar{\mathbf{z}} \in X^m \times Y} Q(\mathbf{z};\bar{\mathbf{z}}) \tag{2.8}$$

measures the accuracy of the approximate solution $\mathbf{z}$ to the saddle point problem (1.8).

However, the saddle point formulation (1.8) of our problem of interest (1.1) may have an unbounded feasible set. We adopt the perturbation-based termination criterion by Monteiro and Svaiter [31,32,33] and propose a modified version of the gap function in (2.8). More specifically, we define

$$g_Y(\mathbf{s},\mathbf{z}) := \sup_{\bar{\mathbf{y}} \in Y} Q(\mathbf{z};\mathbf{x}^*,\bar{\mathbf{y}}) - \langle \mathbf{s}, \bar{\mathbf{y}}\rangle, \tag{2.9}$$

for any closed set $Y \subseteq \mathbb{R}^{md}$, $\mathbf{z} \in X^m \times \mathbb{R}^{md}$ and $\mathbf{s} \in \mathbb{R}^{md}$. If $Y = \mathbb{R}^{md}$, we omit the subscript $Y$ and simply use the notation $g(\mathbf{s},\mathbf{z})$.

This perturbed gap function allows us to bound the objective function value and the feasibility separately. We first define the following terminology.

[1] We can define the norm associated with $X^m$ in a more general way, e.g., $\|\mathbf{x}\|^2 := \sum_{i=1}^m p_i\|x_i\|_{X_i}^2,\ \forall \mathbf{x} = (x_1,\ldots,x_m) \in X^m$, for some $p_i > 0$, $i = 1,\ldots,m$. Accordingly, the prox-function $\mathbf{V}(\cdot,\cdot)$ can be defined as $\mathbf{V}(\mathbf{x},\mathbf{u}) := \sum_{i=1}^m p_i V_i(x_i,u_i),\ \forall \mathbf{x},\mathbf{u} \in X^m$. This setting gives us flexibility to choose $p_i$'s based on the information of individual $X_i$'s, and the possibility to further refine the convergence results.

**Definition 1** A point $\mathbf{x} \in X^m$ is called an $(\epsilon, \delta)$-solution of (1.7) if

$$F(\mathbf{x}) - F(\mathbf{x}^*) \leq \epsilon \text{ and } \|\mathbf{L}\mathbf{x}\| \leq \delta. \tag{2.10}$$

We say that $\mathbf{x}$ has primal residual $\epsilon$ and feasibility residual $\delta$.

Similarly, a stochastic $(\epsilon, \delta)$-solution of (1.7) can be defined as a point $\hat{\mathbf{x}} \in X^m$ s.t. $\mathbb{E}[F(\hat{\mathbf{x}}) - F(\mathbf{x}^*)] \leq \epsilon$ and $\mathbb{E}[\|\mathbf{L}\hat{\mathbf{x}}\|] \leq \delta$ for some $\epsilon, \delta > 0$. Note that for problem (1.7), the feasibility residual measures the disagreement among the local copies $x_i$, for $i \in \mathcal{N}$.

In the following proposition, we adopt a result from [44, Proposition 2.1] to describe the relationship between the perturbed gap function (2.9) and the approximate solutions to problem (1.7). Although the proposition was originally developed for deterministic cases, the extension of this to stochastic cases is straightforward.

**Proposition 1** *For any $Y \subset \mathbb{R}^{md}$ such that $\mathbf{0} \in Y$, if $g_Y(\mathbf{L}\mathbf{x}, \mathbf{z}) \leq \epsilon < \infty$ and $\|\mathbf{L}\mathbf{x}\| \leq \delta$, where $\mathbf{z} = (\mathbf{x}, \mathbf{y}) \in X^m \times \mathbb{R}^{md}$, then $\mathbf{x}$ is an $(\epsilon, \delta)$-solution of (1.7). In particular, when $Y = \mathbb{R}^{md}$, for any $\mathbf{s}$ such that $g(\mathbf{s}, \mathbf{z}) \leq \epsilon < \infty$ and $\|\mathbf{s}\| \leq \delta$, we always have $\mathbf{s} = \mathbf{L}\mathbf{x}$.*

## 3 Decentralized Primal-Dual

In this section, we describe an algorithmic framework for solving the saddle point problem (1.8) in a decentralized fashion. The basic scheme of the decentralized primal-dual method in Algorithm 1 is similar to Chambolle and Pork's primal-dual method in [7]. The primal-dual method in [7] is an efficient and simple method for solving saddle point problems, which can be viewed as a refined version of the primal-dual hybrid gradient method by Arrow et al. [1]. However, its design and analysis is more closely related to a few recent important works which established the $\mathcal{O}(1/k)$ rate of convergence for solving bilinear saddle point problems (e.g., [43, 40, 34, 19]). Recently, Chen, Lan and Ouyang [12] incorporated Bregman distance into the primal-dual method together with an acceleration step. Dang and Lan [13], and Chambolle and Pork [6] discussed improved algorithms for problems with strongly convex primal or dual functions. Randomized versions of the primal-dual method have been discussed by Zhang and Xiao [64], and Dang and Lan [13]. Lan and Zhou [26] revealed some inherent relationship between Nesterov's accelerated gradient method and the primal-dual method, and presented an optimal randomized incremental gradient method.

Our main goals here in this section are to: 1) adapt the primal-dual framework for a decentralized setting; and 2) provide complexity results (number of communication rounds and subgradient computations) separately in terms of primal functional optimality gap and constraint (or consistency) violation. It should be stressed that the main contributions of this paper exist in the development of decentralized communication sliding algorithms (see Section 4 and 5). However, introducing the basic decentralized primal-dual method here will help us better explain these methods and provide us with a certain benchmark in terms of the communication cost.

**Algorithm 1** Decentralized primal-dual

Let $\mathbf{x}^0 = \mathbf{x}^{-1} \in X^m$ and $\mathbf{y}^0 \in \mathbb{R}^{md}$, the nonnegative parameters $\{\alpha_k\}$, $\{\tau_k\}$ and $\{\eta_k\}$, and the weights $\{\theta_k\}$ be given.
**for** $k = 1, \ldots, N$ **do**
Update $\mathbf{z}^k = (\mathbf{x}^k, \mathbf{y}^k)$ according to

$$\tilde{\mathbf{x}}^k = \alpha_k(\mathbf{x}^{k-1} - \mathbf{x}^{k-2}) + \mathbf{x}^{k-1} \tag{3.1}$$

$$\mathbf{y}^k = \operatorname{argmin}_{\mathbf{y} \in \mathbb{R}^{md}} \langle -\mathbf{L}\tilde{\mathbf{x}}^k, \mathbf{y} \rangle + \tfrac{\tau_k}{2}\|\mathbf{y} - \mathbf{y}^{k-1}\|^2 \tag{3.2}$$

$$\mathbf{x}^k = \operatorname{argmin}_{\mathbf{x} \in X^m} \langle \mathbf{L}\mathbf{y}^k, \mathbf{x} \rangle + F(\mathbf{x}) + \eta_k \mathbf{V}(\mathbf{x}^{k-1}, \mathbf{x}) \tag{3.3}$$

**end for**
**return** $\bar{\mathbf{z}}^N = (\sum_{k=1}^N \theta_k)^{-1} \sum_{k=1}^N \theta_k \mathbf{z}^k$.

**Algorithm 2** Decentralized primal-dual update for each agent $i$

Let $x_i^0 = x_i^{-1} \in X_i$ and $y_i^0 \in \mathbb{R}^d$ for $i \in [m]$, the nonnegative parameters $\{\alpha_k\}$, $\{\tau_k\}$ and $\{\eta_k\}$, and the weights $\{\theta_k\}$ be given.
**for** $k = 1, \ldots, N$ **do**
Update $z_i^k = (x_i^k, y_i^k)$ according to

$$\tilde{x}_i^k = \alpha_k(x_i^{k-1} - x_i^{k-2}) + x_i^{k-1} \tag{3.4}$$
$$v_i^k = \textstyle\sum_{j \in N_i} L_{ij} \tilde{x}_j^k \tag{3.5}$$
$$y_i^k = y_i^{k-1} + \tfrac{1}{\tau_k} v_i^k \tag{3.6}$$
$$w_i^k = \textstyle\sum_{j \in N_i} L_{ij} y_j^k \tag{3.7}$$
$$x_i^k = \operatorname{argmin}_{x_i \in X_i} \ \langle w_i^k, x_i \rangle + f_i(x_i) + \eta_k V_i(x_i^{k-1}, x_i) \tag{3.8}$$

**end for**
**return** $\bar{\mathbf{z}}^N = (\sum_{k=1}^N \theta_k)^{-1} \sum_{k=1}^N \theta_k \mathbf{z}^k$

### 3.1 The Algorithm

The primal-dual algorithm in Algorithm 1 can be decentralized due to the structure of the Laplacian $\mathbf{L}$. Recalling that $\mathbf{x} = (x_1, \ldots, x_m)$ and $\mathbf{y} = (y_1, \ldots, y_m)$, each agent $i$'s local update rule can be separately written as in Algorithm 2. Each agent $i$ maintains two local sequences, namely, the primal estimates $\{x_i^k\}$ and the dual variables $\{y_i^k\}$. The element $x_i^k$ can be seen as agent $i$'s estimate of the decision variable $x$ at time $k$, while $y_i^k$ is a subvector of all dual variables $\mathbf{y}^k$ associated with the agent $i$'s consistency constraints with its neighbors.

More specifically, each primal estimate $x_i^0$ is locally initialized from some arbitrary point in $X_i$, and $x_i^{-1}$ is also set to be the same value. At each time step $k \geq 1$, each agent $i \in \mathcal{N}$ computes a local prediction $\tilde{x}_i^k$ using the two previous primal estimates (ref. (3.4)), and broadcasts this to all of the nodes in its neighborhood, i.e., to all agents $j \in N_i$. In (3.5)-(3.6), each agent $i$ calculates the neighborhood disagreement $v_i^k$ using the messages received from agents in $N_i$, and updates the dual subvector $y_i^k$. Then, another round of communication occurs in (3.7) to broadcast this updated dual variables and calculate $w_i^k$. Therefore, each iteration $k$ involves two communication rounds, one for the primal estimates and the other for the dual variables. Lastly, each agent $i$ solves the proximal projection subproblem (3.8). Note that the description of the algorithm is only conceptual at this moment since we have not specified the parameters $\{\alpha_k\}$, $\{\tau_k\}$, $\{\eta_k\}$ and $\{\theta_k\}$ yet. We will later instantiate this generic algorithm when we state its convergence properties.

### 3.2 Convergence of the Decentralized Primal-dual Method

For the sake of simplicity, we focus only on the case when $f_i$'s are general convex functions in this section. We leave the discussion about the case when $f_i$'s are strongly convex later in Sections 4 and 5 for decentralized communication sliding algorithms.

In the following lemma, we present estimates on the gap function defined in (2.7) together with conditions on the parameters $\{\alpha_k\}$, $\{\tau_k\}$, $\{\eta_k\}$ and $\{\theta_k\}$, which will be used to provide the rate of convergence for the decentralized primal-dual method. The proof of this lemma can be found in Section 6.

**Lemma 1** *Let the iterates $\mathbf{z}^k = (\mathbf{x}^k, \mathbf{y}^k)$, $k = 1, \ldots, N$ be generated by Algorithm 1 and $\bar{\mathbf{z}}^N$ be defined as $\bar{\mathbf{z}}^N := \left(\sum_{k=1}^N \theta_k\right)^{-1} \sum_{k=1}^N \theta_k \mathbf{z}^k$. Assume that the parameters $\{\alpha_k\}$, $\{\tau_k\}$, $\{\eta_k\}$ and $\{\theta_k\}$ in Algorithm 1 satisfy*

$$\theta_k \eta_k \leq \theta_{k-1} \eta_{k-1}, \quad k = 2, \ldots, N, \tag{3.9}$$
$$\alpha_k \theta_k = \theta_{k-1}, \quad k = 2, \ldots, N, \tag{3.10}$$
$$\theta_k \tau_k \leq \theta_{k-1} \tau_{k-1}, \quad k = 2, \ldots, N, \tag{3.11}$$
$$\alpha_k \|\mathbf{L}\|^2 \leq \eta_{k-1} \tau_k, \quad k = 2, \ldots, N, \tag{3.12}$$
$$\theta_1 \tau_1 = \theta_N \tau_N, \tag{3.13}$$
$$\theta_N \|\mathbf{L}\|^2 \leq \theta_1 \tau_1 \eta_N. \tag{3.14}$$

*Then, for any* $\mathbf{z} := (\mathbf{x}, \mathbf{y}) \in X^m \times \mathbb{R}^{md}$*, we have*

$$Q(\bar{\mathbf{z}}^N; \mathbf{z}) \leq \left(\sum_{k=1}^N \theta_k\right)^{-1} \left[\theta_1 \eta_1 \mathbf{V}(\mathbf{x}^0, \mathbf{x}) + \tfrac{\theta_1 \tau_1}{2}\|\mathbf{y}^0\|^2 + \langle \mathbf{s}, \mathbf{y}\rangle\right], \tag{3.15}$$

*where* $Q$ *is defined in* (2.7) *and* $\mathbf{s}$ *is defined as*

$$\mathbf{s} := \theta_N \mathbf{L}(\mathbf{x}^N - \mathbf{x}^{N-1}) + \theta_1 \tau_1 (\mathbf{y}^N - \mathbf{y}^0). \tag{3.16}$$

*Furthermore, for any saddle point* $(\mathbf{x}^*, \mathbf{y}^*)$ *of* (1.8)*, we have*

$$\tfrac{\theta_N}{2}\left(1 - \tfrac{\|\mathbf{L}\|^2}{\eta_N \tau_N}\right) \max\{\eta_N \|\mathbf{x}^{N-1} - \mathbf{x}^N\|^2, \tau_N \|\mathbf{y}^* - \mathbf{y}^N\|^2\} \leq \theta_1 \eta_1 \mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \tfrac{\theta_1 \tau_1}{2}\|\mathbf{y}^* - \mathbf{y}^0\|^2. \tag{3.17}$$

In the following theorem, we provide a specific selection of $\{\alpha_k\}$, $\{\tau_k\}$, $\{\eta_k\}$ and $\{\theta_k\}$ satisfying (3.9)-(3.14). Using Lemma 1 and Proposition 1, we also establish the complexity of the decentralized primal-dual method for computing an $(\epsilon, \delta)$-solution of problem (1.7).

**Theorem 1** *Let* $\mathbf{x}^*$ *be a saddle point of* (1.7)*, and suppose that* $\{\alpha_k\}$*,* $\{\tau_k\}$*,* $\{\eta_k\}$ *and* $\{\theta_k\}$ *are set to*

$$\alpha_k = \theta_k = 1, \ \eta_k = 2\|\mathbf{L}\|, \ \text{and } \tau_k = \|\mathbf{L}\|, \quad \forall k = 1, \ldots, N. \tag{3.18}$$

*Then, for any* $N \geq 1$*, we have*

$$F(\bar{\mathbf{x}}^N) - F(\mathbf{x}^*) \leq \tfrac{\|\mathbf{L}\|}{N}\left[2\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \tfrac{1}{2}\|\mathbf{y}^0\|^2\right] \tag{3.19}$$

*and*

$$\|\mathbf{L}\bar{\mathbf{x}}^N\| \leq \tfrac{2\|\mathbf{L}\|}{N}\left[3\sqrt{\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*)} + 2\|\mathbf{y}^* - \mathbf{y}^0\|\right], \tag{3.20}$$

*where* $\bar{\mathbf{x}}^N = \frac{1}{N}\sum_{k=1}^N \mathbf{x}^k$*.*

*Proof* It is easy to check that (3.18) satisfies conditions (3.9)-(3.14). Therefore, by plugging these values in (3.15), we have

$$Q(\bar{\mathbf{z}}^N; \mathbf{x}^*, \mathbf{y}) \leq \tfrac{1}{N}\left[2\|\mathbf{L}\|\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \tfrac{\|\mathbf{L}\|}{2}\|\mathbf{y}^0\|^2\right] + \tfrac{1}{N}\langle \mathbf{s}, \mathbf{y}\rangle. \tag{3.21}$$

Letting $\mathbf{s}^N := \frac{1}{N}\mathbf{s}$, then from (3.16) and (3.17) we have

$$\begin{aligned}\|\mathbf{s}^N\| &\leq \tfrac{\|\mathbf{L}\|}{N}\left[\|\mathbf{x}^N - \mathbf{x}^{N-1}\| + \|\mathbf{y}^N - \mathbf{y}^*\| + \|\mathbf{y}^* - \mathbf{y}^0\|\right] \\ &\leq \tfrac{\|\mathbf{L}\|}{N}\left[3\sqrt{4\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \|\mathbf{y}^* - \mathbf{y}^0\|^2} + \|\mathbf{y}^* - \mathbf{y}^0\|\right].\end{aligned}$$

Furthermore, by (3.21) we have

$$g(\mathbf{s}^N, \bar{\mathbf{z}}^N) \leq \tfrac{\|\mathbf{L}\|}{N}\left[2\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \tfrac{1}{2}\|\mathbf{y}^0\|^2\right].$$

The results in (3.19) and (3.20) then immediately follow from Proposition 1 and the above two inequalities.

From (3.19)-(3.20), we can see that the complexity of decentralized primal-dual method for computing an $(\epsilon, \delta)$-solution is $\mathcal{O}(1/\epsilon)$ for the primal functional optimality and $\mathcal{O}(1/\delta)$ for the constraint violation. Since each iteration involves a constant number of communication rounds, the number of inter-node communications required is also in the same order.

## 4 Decentralized Communication Sliding

In this section, we present a new decentralized primal-dual type method, namely, the decentralized communication sliding (DCS) method for the case when the primal subproblem (3.8) is not easy to solve. We show that one can still maintain the same number of inter-node communications even when the subproblem (3.8) is approximately solved through an iterative subgradient descent procedure, and that the total number of required subgradient evaluations is comparable to centralized mirror descent methods. Throughout this section, we consider the deterministic case where exact subgradients of $f_i$'s are available.

**Algorithm 3** Decentralized Communication Sliding (DCS)

Let $x_i^0 = x_i^{-1} = \hat{x}_i^0 \in X_i$, $y_i^0 \in \mathbb{R}^d$ for $i \in [m]$ and the nonnegative parameters $\{\alpha_k\}$, $\{\tau_k\}$, $\{\eta_k\}$ and $\{T_k\}$ be given.
**for** $k = 1, \ldots, N$ **do**
Update $\mathbf{z}^k = (\hat{\mathbf{x}}^k, \mathbf{y}^k)$ according to

$$\tilde{x}_i^k = \alpha_k(\hat{x}_i^{k-1} - x_i^{k-2}) + x_i^{k-1} \tag{4.1}$$
$$v_i^k = \textstyle\sum_{j \in N_i} L_{ij} \tilde{x}_j^k \tag{4.2}$$
$$y_i^k = \operatorname{argmin}_{y_i \in \mathbb{R}^d} \langle -v_i^k, y_i \rangle + \tfrac{\tau_k}{2} \|y_i - y_i^{k-1}\|^2 = y_i^{k-1} + \tfrac{1}{\tau_k} v_i^k \tag{4.3}$$
$$w_i^k = \textstyle\sum_{j \in N_i} L_{ij} y_j^k \tag{4.4}$$
$$(x_i^k, \hat{x}_i^k) = \mathrm{CS}(f_i, X_i, V_i, T_k, \eta_k, w_i^k, x_i^{k-1}) \tag{4.5}$$

**end for**
**return** $z_i^N = (\hat{x_i}^N, y_i^N)$

The CS (Communication-Sliding) procedure called at (4.5) is stated as follows.
**procedure:** $(x, \hat{x}) = \mathrm{CS}(\phi, U, V, T, \eta, w, x)$
Let $u^0 = \hat{u}^0 = x$ and the parameters $\{\beta_t\}$ and $\{\lambda_t\}$ be given.
**for** $t = 1, \ldots, T$ **do**

$$h^{t-1} = \phi'(u^{t-1}) \in \partial\phi(u^{t-1}) \tag{4.6}$$
$$u^t = \operatorname{argmin}_{u \in U} \left[\langle w + h^{t-1}, u\rangle + \eta V(x, u) + \eta\beta_t V(u^{t-1}, u)\right] \tag{4.7}$$

**end for**
Set

$$\hat{u}^T := \left(\textstyle\sum_{t=1}^T \lambda_t\right)^{-1} \sum_{t=1}^T \lambda_t u^t. \tag{4.8}$$

Set $x = u^T$ and $\hat{x} = \hat{u}^T$.
**end procedure**

## 4.1 The DCS Algorithm

We formally describe our DCS algorithm in Algorithm 3. We say that an outer iteration of the DCS algorithm, which we call the outer-loop, occurs whenever the index $k$ in Algorithm 3 is incremented by 1. Since the subproblems are solved inexactly, the outer-loop of the primal-dual algorithm also needs to be modified in order to attain the best possible rate of convergence. In particular, in addition to the primal estimate $\{x_i^k\}$, we let each agent $i$ maintain another primal sequence $\{\hat{x}_i^k\}$ (cf. the definition of $\tilde{x}_i^k$ in (4.1)), which will later play a crucial role in the development and convergence proof of the algorithm. Observe that the DCS method, in spirit, has been inspired by some of our recent work on gradient sliding [24]. However, the gradient sliding method in [24] focuses on how to save gradient evaluations for solving certain structured convex optimization problems, rather than how to save communication rounds for decentralized optimization, and its algorithmic scheme is also quite different from the DCS method.

The steps (4.1)-(4.4) are similar to those in Algorithm 2 except that the local prediction $\tilde{x}_i^k$ in (4.1) is computed using the two previous primal estimates $\hat{x}_i^{k-1}$ and $x_i^{k-1}$. The CS procedure in (4.5), which we call the inner loop, solves the subproblem (3.8) iteratively for $T_k$ iterations. Each inner loop iteration consists of the computation of the subgradient $f_i(u^{t-1})$ in (4.6) and the solution of the projection subproblem in (4.7), which is assumed to be relatively easy to solve. Note that the description of the algorithm is only conceptual at this moment since we have not specified the parameters $\{\alpha_k\}$, $\{\eta_k\}$, $\{\tau_k\}$, $\{T_k\}$ $\{\beta_t\}$ and $\{\lambda_t\}$ yet. We will later instantiate this generic algorithm when we state its convergence properties.

A few remarks about this algorithm are in order. Firstly, a critical difference of this routine compared to the exact version (Algorithm 2) is that one needs to compute a pair of approximate solutions $x_i^k$ and $\hat{x}_i^k$. While both $x_i^k$ and $\hat{x}_i^k$ can be seen as agent $i$'s estimate of the decision variable $x$ at time $k$, $x_i^k$ will be used to define the subproblem (4.7) for the next call to the CS procedure and $\hat{x}_i^k$ will be used to produce a weighted sum of all the inner loop iterates. Secondly, since the same $w_i^k$ has been used throughout the $T_k$ iterations of the CS procedure, no additional communications of

the dual variables are required when performing the subgradient projection step (4.7) for $T_k$ times. This differs from the accelerated gradient methods in [11,22] where the number of inter-node communications at each iteration $k$ increase linearly or sublinearly in the order of $k$.

Note that the results of the CS procedure at iteration $k$ for agents $i \in \mathcal{N}$ collectively generate a pair of approximate solutions $\hat{\mathbf{x}}^k = (\hat{x}_1^k, \ldots, \hat{x}_m^k)$ and $\mathbf{x}^k = (x_1^k, \ldots, x_m^k)$ to the proximal projection subproblem (3.3). For later convenience, we refer to the subproblem at iteration $k$ as $\Phi^k(\mathbf{x})$, i.e.,

$$\operatorname{argmin}_{\mathbf{x}\in X^m} \left\{ \Phi^k(\mathbf{x}) := \langle \mathbf{L}\mathbf{y}^k, \mathbf{x}\rangle + F(\mathbf{x}) + \eta_k \mathbf{V}(\mathbf{x}^{k-1}, \mathbf{x}) \right\}. \tag{4.9}$$

## 4.2 Convergence of DCS on General Convex Functions

We now establish the main convergence properties of the DCS algorithm. More specifically, we provide in Lemma 2 an estimate on the gap function defined in (2.7) together with stepsize policies which work for the general nonsmooth convex case with $\mu = 0$ (cf. (1.2)). The proof of this lemma can be found in Section 6.

**Lemma 2** *Let the iterates $(\hat{\mathbf{x}}^k, \mathbf{y}^k)$, $k = 1, \ldots, N$ be generated by Algorithm 3 and $\hat{\mathbf{z}}^N$ be defined as $\hat{\mathbf{z}}^N := \left(\sum_{k=1}^N \theta_k\right)^{-1} \sum_{k=1}^N \theta_k(\hat{\mathbf{x}}^k, \mathbf{y}^k)$. Assume that the objective $f_i$, $i = 1, \ldots, m$, are general nonsmooth convex functions, i.e., $\mu = 0$ and $M > 0$. Let the parameters $\{\alpha_k\}$, $\{\theta_k\}$, $\{\eta_k\}$, $\{\tau_k\}$ and $\{T_k\}$ in Algorithm 3 satisfy* (3.10)-(3.14) *and*

$$\theta_k \tfrac{(T_k+1)(T_k+2)\eta_k}{T_k(T_k+3)} \le \theta_{k-1} \tfrac{(T_{k-1}+1)(T_{k-1}+2)\eta_{k-1}}{T_{k-1}(T_{k-1}+3)}, \quad k = 2, \ldots, N. \tag{4.10}$$

*Let the parameters $\{\lambda_t\}$ and $\{\beta_t\}$ in the CS procedure of Algorithm 3 be set to*

$$\lambda_t = t + 1, \quad \beta_t = \tfrac{t}{2}, \quad \forall t \ge 1. \tag{4.11}$$

*Then, we have for all $\mathbf{z} \in X^m \times \mathbb{R}^{md}$,*

$$Q(\hat{\mathbf{z}}^N; \mathbf{z}) \le \left(\textstyle\sum_{k=1}^N \theta_k\right)^{-1} \left[ \tfrac{(T_1+1)(T_1+2)\theta_1\eta_1}{T_1(T_1+3)} \mathbf{V}(\mathbf{x}^0, \mathbf{x}) + \tfrac{\theta_1\tau_1}{2}\|\mathbf{y}^0\|^2 + \langle \hat{\mathbf{s}}, \mathbf{y}\rangle + \textstyle\sum_{k=1}^N \tfrac{4mM^2\theta_k}{(T_k+3)\eta_k} \right], \tag{4.12}$$

*where $\hat{\mathbf{s}} := \theta_N \mathbf{L}(\hat{\mathbf{x}}^N - \mathbf{x}^{N-1}) + \theta_1\tau_1(\mathbf{y}^N - \mathbf{y}^0)$ and $Q$ is defined in* (2.7)*. Furthermore, for any saddle point $(\mathbf{x}^*, \mathbf{y}^*)$ of* (1.8)*, we have*

$$\begin{aligned} &\tfrac{\theta_N}{2}\left(1 - \tfrac{\|\mathbf{L}\|^2}{\eta_N\tau_N}\right) \max\{\eta_N\|\hat{\mathbf{x}}^N - \mathbf{x}^{N-1}\|^2, \tau_N\|\mathbf{y}^* - \mathbf{y}^N\|^2\} \\ &\qquad \le \tfrac{(T_1+1)(T_1+2)\theta_1\eta_1}{T_1(T_1+3)} \mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \tfrac{\theta_1\tau_1}{2}\|\mathbf{y}^* - \mathbf{y}^0\|^2 + \textstyle\sum_{k=1}^N \tfrac{4mM^2\theta_k}{\eta_k(T_k+3)}. \end{aligned} \tag{4.13}$$

In the following theorem, we provide a specific selection of $\{\alpha_k\}$, $\{\theta_k\}$, $\{\eta_k\}$, $\{\tau_k\}$ and $\{T_k\}$ satisfying (3.10)-(3.14) and (4.10). Using Lemma 2 and Proposition 1, we also establish the complexity of the DCS method for computing an $(\epsilon, \delta)$-solution of problem (1.7) when the objective functions are general convex.

**Theorem 2** *Let $\mathbf{x}^*$ be an optimal solution of* (1.7)*, the parameters $\{\lambda_t\}$ and $\{\beta_t\}$ in the CS procedure of Algorithm 3 be set to* (4.11)*, and suppose that $\{\alpha_k\}$, $\{\theta_k\}$, $\{\eta_k\}$, $\{\tau_k\}$ and $\{T_k\}$ are set to*

$$\alpha_k = \theta_k = 1, \ \eta_k = 2\|\mathbf{L}\|, \ \tau_k = \|\mathbf{L}\|, \ \text{and } T_k = \left\lceil \tfrac{mM^2N}{\|\mathbf{L}\|^2\tilde{D}} \right\rceil, \quad \forall k = 1, \ldots, N, \tag{4.14}$$

*for some $\tilde{D} > 0$. Then, for any $N \ge 1$, we have*

$$F(\hat{\mathbf{x}}^N) - F(\mathbf{x}^*) \le \tfrac{\|\mathbf{L}\|}{N}\left[3\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \tfrac{1}{2}\|\mathbf{y}^0\|^2 + 2\tilde{D}\right] \tag{4.15}$$

*and*

$$\|\mathbf{L}\hat{\mathbf{x}}^N\| \le \tfrac{\|\mathbf{L}\|}{N}\left[3\sqrt{6\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + 4\tilde{D}} + 4\|\mathbf{y}^* - \mathbf{y}^0\|\right], \tag{4.16}$$

*where $\hat{\mathbf{x}}^N = \frac{1}{N}\sum_{k=1}^N \hat{\mathbf{x}}^k$.*

*Proof* It is easy to check that (4.14) satisfies conditions (3.10)-(3.14) and (4.10). Particularly,

$$\tfrac{(T_1+1)(T_1+2)}{T_1(T_1+3)} = 1 + \tfrac{2}{T_1^2+3T_1} \le \tfrac{3}{2}.$$

Therefore, by plugging in these values to (4.12), we have

$$Q(\hat{\mathbf{z}}^N; \mathbf{x}^*, \mathbf{y}) \le \tfrac{\|\mathbf{L}\|}{N}\left[3\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \tfrac{1}{2}\|\mathbf{y}^0\|^2 + 2\tilde{D}\right] + \tfrac{1}{N}\langle \hat{\mathbf{s}}, \mathbf{y}\rangle. \tag{4.17}$$

Letting $\hat{\mathbf{s}}^N = \frac{1}{N}\hat{\mathbf{s}}$, then from (4.13), we have

$$\begin{aligned}\|\hat{\mathbf{s}}^N\| &\le \tfrac{\|\mathbf{L}\|}{N}\left[\|\hat{\mathbf{x}}^N - \mathbf{x}^{N-1}\| + \|\mathbf{y}^N - \mathbf{y}^*\| + \|\mathbf{y}^* - \mathbf{y}^0\|\right] \\ &\le \tfrac{\|\mathbf{L}\|}{N}\left[3\sqrt{6\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \|\mathbf{y}^* - \mathbf{y}^0\|^2 + 4\tilde{D}} + \|\mathbf{y}^* - \mathbf{y}^0\|\right].\end{aligned}$$

Furthermore, by (4.17), we have

$$g(\hat{\mathbf{s}}^N, \hat{\mathbf{z}}^N) \le \tfrac{\|\mathbf{L}\|}{N}\left[3\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \tfrac{1}{2}\|\mathbf{y}^0\|^2 + 2\tilde{D}\right].$$

Applying Proposition 1 to the above two inequalities, the results in (4.15) and (4.16) follow immediately.

We now make some remarks about the results obtained in Theorem 2. Firstly, even though one can choose any $\tilde{D} > 0$ (e.g., $\tilde{D} = 1$) in (4.14), the best selection of $\tilde{D}$ would be $\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*)$ so that the first and third terms in (4.17) are about the same order. In practice, if there exists an estimate $\mathcal{D}_{X^m} > 0$ s.t.

$$\mathbf{V}(\mathbf{x}_1, \mathbf{x}_2) \le \mathcal{D}_{X^m}^2, \ \forall \mathbf{x}_1, \mathbf{x}_2 \in X^m, \tag{4.18}$$

then we can set $\tilde{D} = \mathcal{D}_{X^m}^2$.

Secondly, the complexity of the DCS method directly follows from (4.15) and (4.16). For simplicity, let us assume that $X$ is bounded, $\tilde{D} = \mathcal{D}_{X^m}^2$ and $\mathbf{y}^0 = \mathbf{0}$. We can see that the total number of inter-node communication rounds and intra-node subgradient evaluations required by each agent for finding an $(\epsilon, \delta)$-solution of (1.7) can be bounded by

$$\mathcal{O}\left\{\|\mathbf{L}\| \max\left(\tfrac{\mathcal{D}_{X^m}^2}{\epsilon}, \tfrac{\mathcal{D}_{X^m} + \|\mathbf{y}^*\|}{\delta}\right)\right\} \ \text{ and } \ \mathcal{O}\left\{mM^2 \max\left(\tfrac{\mathcal{D}_{X^m}^2}{\epsilon^2}, \tfrac{\mathcal{D}_{X^m}^2 + \|\mathbf{y}^*\|^2}{\mathcal{D}_{X^m}^2 \delta^2}\right)\right\}, \tag{4.19}$$

respectively. In particular, if $\epsilon$ and $\delta$ satisfy

$$\tfrac{\epsilon}{\delta} \le \tfrac{\mathcal{D}_{X^m}^2}{\mathcal{D}_{X^m} + \|\mathbf{y}^*\|}, \tag{4.20}$$

then the previous two complexity bounds in (4.19), respectively, reduce to

$$\mathcal{O}\left\{\tfrac{\|\mathbf{L}\|\mathcal{D}_{X^m}^2}{\epsilon}\right\} \ \text{ and } \ \mathcal{O}\left\{\tfrac{mM^2\mathcal{D}_{X^m}^2}{\epsilon^2}\right\}. \tag{4.21}$$

Thirdly, it is interesting to compare DCS with the centralized mirror descent method [42] applied to (1.1). In the worst case, the Lipschitz constant of $f$ in (1.1) can be bounded by $M_f \le mM$, and each iteration of the method will incur $m$ subgradient evaluations. Hence, the total number of subgradient evaluations performed by the mirror descent method for finding an $\epsilon$-solution of (1.1), i.e., a point $\bar{x} \in X$ such that $f(\bar{x}) - f^* \le \epsilon$, can be bounded by

$$\mathcal{O}\left\{\tfrac{m^3M^2\mathcal{D}_X^2}{\epsilon^2}\right\}, \tag{4.22}$$

where $\mathcal{D}_X^2$ characterizes the diameter of $X$, i.e., $\mathcal{D}_X^2 := \max_{x_1, x_2 \in X} V(x_1, x_2)$. Noting that $\mathcal{D}_X^2/\mathcal{D}_{X^m}^2 = \mathcal{O}(1/m)$, and that the second bound in (4.21) states only the number of subgradient evaluations for each agent in the DCS method, we conclude that the total number of subgradient evaluations performed by DCS is comparable to the classic mirror descent method as long as (4.20) holds and hence not improvable in general.

4.3 Boundedness of $\|\mathbf{y}^*\|$

In this subsection, we will provide a bound on the optimal dual multiplier $\mathbf{y}^*$. By doing so, we show that the complexity of DCS algorithm (as well as the stochastic DCS algorithm in Section 5) only depends on the parameters for the primal problem along with the smallest singular value of $\mathbf{L}$ and the initial point $\mathbf{y}^0$, even though these algorithms are intrinsically primal-dual type methods.

**Theorem 3** *Let $\mathbf{x}^*$ be an optimal solution of (1.7). Then there exists an optimal dual multiplier $\mathbf{y}^*$ for (1.8) s.t.*

$$\|\mathbf{y}^*\| \leq \tfrac{\sqrt{m}M}{\tilde{\sigma}_{min}(\mathbf{L})}, \tag{4.23}$$

*where $\tilde{\sigma}_{min}(\mathbf{L})$ denotes the smallest nonzero singular value of $\mathbf{L}$.*

*Proof* Since we only relax the linear constraints in problem (1.7) to obtain the Lagrange dual problem (1.8), it follows from the strong Lagrange duality and the existence of $\mathbf{x}^*$ to (1.7) that an optimal dual multiplier $\mathbf{y}^*$ for problem (1.8) must exist. It is clear that

$$\mathbf{y}^* = \mathbf{y}^*_N + \mathbf{y}^*_C,$$

where $\mathbf{y}^*_N$ and $\mathbf{y}^*_C$ denote the projections of $\mathbf{y}^*$ over the null space and the column space of $\mathbf{L}^T$, respectively.

We consider two cases. Case 1) $\mathbf{y}^*_C = \mathbf{0}$. Since $\mathbf{y}^*_N$ belongs to the null space of $\mathbf{L}^T$, $\mathbf{L}^T\mathbf{y}^* = \mathbf{L}^T\mathbf{y}^*_N = \mathbf{0}$, which implies that for any $c \in \mathbb{R}$, $c\mathbf{y}^*$ is also an optimal dual multiplier of (1.8). Therefore, (4.23) clearly holds, because we can scale $\mathbf{y}^*$ to an arbitrarily small vector.

Case 2) $\mathbf{y}^*_C \neq \mathbf{0}$. Using the fact that $\mathbf{L}^T\mathbf{y}^* = \mathbf{L}^T\mathbf{y}^*_C$ and the definition of a saddle point of (1.8), we conclude that $\mathbf{y}^*_C$ is also an optimal dual multiplier of (1.8). Since $\mathbf{y}^*_C$ is in the column space of $\mathbf{L}$, we have

$$\|\mathbf{L}^T\mathbf{y}^*_C\|^2 = (\mathbf{y}^*_C)^T\mathbf{L}\mathbf{L}^T\mathbf{y}^*_C = (\mathbf{y}^*_C)^T\mathbf{U}^T\mathbf{\Lambda}\mathbf{U}\mathbf{y}^*_C \geq \tilde{\lambda}_{\min}(\mathbf{L}\mathbf{L}^T)\|\mathbf{U}\mathbf{y}^*_C\|^2 = \tilde{\sigma}^2_{\min}(\mathbf{L})\|\mathbf{y}^*_C\|^2,$$

where $\mathbf{U}$ is an orthonormal matrix whose rows consist of the eigenvectors of $\mathbf{L}\mathbf{L}^T$, $\mathbf{\Lambda}$ is the diagonal matrix whose diagonal elements are the corresponding eigenvalues, $\tilde{\lambda}_{min}(\mathbf{L}\mathbf{L}^T)$ denotes the smallest nonzero eigenvalue of $\mathbf{L}\mathbf{L}^T$, and $\tilde{\sigma}_{min}(\mathbf{L})$ denotes the smallest nonzero singular value of $\mathbf{L}$. In particular,

$$\|\mathbf{y}^*_C\| \leq \tfrac{\|\mathbf{L}^T\mathbf{y}^*_C\|}{\tilde{\sigma}_{\min}(\mathbf{L})}. \tag{4.24}$$

Moreover, if we denote the saddle point problem defined in (1.8) as follows:

$$\mathcal{L}(\mathbf{x}, \mathbf{y}) := F(\mathbf{x}) + \langle \mathbf{L}\mathbf{x}, \mathbf{y}\rangle.$$

By the definition of a saddle point of (1.8), we have $\mathcal{L}(\mathbf{x}^*, \mathbf{y}^*_C) \leq \mathcal{L}(\mathbf{x}, \mathbf{y}^*_C)$, i.e.,

$$F(\mathbf{x}^*) - F(\mathbf{x}) \leq \langle -\mathbf{L}^T\mathbf{y}^*_C, \mathbf{x} - \mathbf{x}^*\rangle.$$

Hence, from the definition of subgradients, we conclude that $-\mathbf{L}^T\mathbf{y}^*_C \in \partial F(\mathbf{x}^*)$, which together with the fact that $F(\cdot)$ is Lipschitz continuous implies that

$$\|\mathbf{L}^T\mathbf{y}^*_C\| = \|\textstyle\sum_{i=1}^m f_i'(x_i^*)\| \leq \sqrt{m}M.$$

Our result in (4.23) follows immediately from the above relation, (4.24) and the fact that $\mathbf{y}^*_C$ is also an optimal dual multiplier of (1.8).

Observe that our bound for the dual multiplier $\mathbf{y}^*$ in (4.23) contains only the primal information. Given an initial dual multiplier $\mathbf{y}^0$, this result can be used to provide an upper bound on $\|\mathbf{y}^0 - \mathbf{y}^*\|$ in Theorems 1-6 throughout this paper. Note also that we can assume $\mathbf{y}^0 = 0$ to simplify these complexity bounds.

4.4 Convergence of DCS on Strongly Convex Functions

In this subsection, we assume that the objective functions $f_i$'s are strongly convex (i.e., $\mu > 0$ (1.2)). In order to take advantage of the strong convexity of the objective functions, we assume that the prox-functions $V_i(\cdot,\cdot)$, $i = 1,\ldots,m$, (cf. (2.2)) are growing quadratically with the *quadratic growth constant* $\mathcal{C}$, i.e., there exists a constant $\mathcal{C} > 0$ such that

$$V_i(x_i, u_i) \le \tfrac{\mathcal{C}}{2}\|x_i - u_i\|_{X_i}^2, \quad \forall x_i, u_i \in X_i,\ i = 1,\ldots,m. \tag{4.25}$$

By (2.3), we must have $\mathcal{C} \ge 1$.

We next provide in Lemma 3 an estimate on the gap function defined in (2.7) together with stepsize policies which work for the strongly convex case. The proof of this lemma can be found in Section 6.

**Lemma 3** *Let the iterates $(\hat{\mathbf{x}}^k, \mathbf{y}^k)$, $k = 1,\ldots,N$ be generated by Algorithm 3 and $\hat{\mathbf{z}}^N$ be defined as $\hat{\mathbf{z}}^N := \left(\sum_{k=1}^N \theta_k\right)^{-1} \sum_{k=1}^N \theta_k(\hat{\mathbf{x}}^k, \mathbf{y}^k)$. Assume the objective $f_i$, $i = 1,\ldots,m$ are strongly convex functions, i.e., $\mu, M > 0$. Let the parameters $\{\alpha_k\}$, $\{\theta_k\}$, $\{\eta_k\}$ and $\{\tau_k\}$ in Algorithm 3 satisfy* (3.10)-(3.14) *and*

$$\theta_k\eta_k \le \theta_{k-1}(\mu/\mathcal{C} + \eta_{k-1}), \quad k = 2,\ldots,N. \tag{4.26}$$

*Let the parameters $\{\lambda_t\}$ and $\{\beta_t\}$ in the CS procedure of Algorithm 3 be set to*

$$\lambda_t = t, \quad \beta_t^{(k)} = \tfrac{(t+1)\mu}{2\eta_k\mathcal{C}} + \tfrac{t-1}{2}, \quad \forall t \ge 1. \tag{4.27}$$

*Then, we have for all $\mathbf{z} \in X^m \times \mathbb{R}^{md}$*

$$Q(\hat{\mathbf{z}}^N; \mathbf{z}) \le \left(\textstyle\sum_{k=1}^N \theta_k\right)^{-1} \left[\theta_1\eta_1 \mathbf{V}(\mathbf{x}^0, \mathbf{x}) + \tfrac{\theta_1\tau_1}{2}\|\mathbf{y}^0\|^2 + \langle \hat{\mathbf{s}}, \mathbf{y}\rangle + \textstyle\sum_{k=1}^N \sum_{t=1}^{T_k} \frac{2mM^2\theta_k}{T_k(T_k+1)} \frac{t}{(t+1)\mu/\mathcal{C} + (t-1)\eta_k}\right], \tag{4.28}$$

*where $\hat{\mathbf{s}} := \theta_N \mathbf{L}(\hat{\mathbf{x}}^N - \mathbf{x}^{N-1}) + \theta_1\tau_1(\mathbf{y}^N - \mathbf{y}^0)$ and $Q$ is defined in* (2.7). *Furthermore, for any saddle point $(\mathbf{x}^*, \mathbf{y}^*)$ of* (1.8), *we have*

$$\begin{aligned} &\tfrac{\theta_N}{2}\left(1 - \tfrac{\|\mathbf{L}\|^2}{\eta_N\tau_N}\right) \max\{\eta_N\|\hat{\mathbf{x}}^N - \mathbf{x}^{N-1}\|^2, \tau_N\|\mathbf{y}^* - \mathbf{y}^N\|^2\} \\ &\qquad \le \theta_1\eta_1 \mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \tfrac{\theta_1\tau_1}{2}\|\mathbf{y}^* - \mathbf{y}^0\|^2 + \textstyle\sum_{k=1}^N \sum_{t=1}^{T_k} \frac{2mM^2\theta_k}{T_k(T_k+1)} \frac{t}{(t+1)\mu/\mathcal{C} + (t-1)\eta_k}. \end{aligned} \tag{4.29}$$

In the following theorem, we provide a specific selection of $\{\alpha_k\}$, $\{\theta_k\}$, $\{\eta_k\}$, $\{\tau_k\}$ and $\{T_k\}$ satisfying (3.10)-(3.14) and (4.26). Also, by using Lemma 3 and Proposition 1, we establish the complexity of the DCS method for computing an $(\epsilon, \delta)$-solution of problem (1.7) when the objective functions are strongly convex. The choice of variable stepsizes rather than using constant stepsizes will accelerate its convergence rate.

**Theorem 4** *Let $\mathbf{x}^*$ be an optimal solution of* (1.7), *the parameters $\{\lambda_t\}$ and $\{\beta_t\}$ in the CS procedure of Algorithm 3 be set to* (4.27) *and suppose that $\{\alpha_k\}$, $\{\theta_k\}$, $\{\eta_k\}$, $\{\tau_k\}$ and $\{T_k\}$ are set to*

$$\alpha_k = \tfrac{k}{k+1},\ \theta_k = k+1,\ \eta_k = \tfrac{k\mu}{2\mathcal{C}},\ \tau_k = \tfrac{4\|\mathbf{L}\|^2\mathcal{C}}{(k+1)\mu},\ \text{and } T_k = \left\lceil \sqrt{\tfrac{2m}{\tilde{D}}}\tfrac{\mathcal{C}MN}{\mu} \max\left\{\sqrt{\tfrac{2m}{\tilde{D}}}\tfrac{4\mathcal{C}M}{\mu}, 1\right\}\right\rceil, \tag{4.30}$$

*$\forall k = 1,\ldots,N$, for some $\tilde{D} > 0$. Then, for any $N \ge 2$, we have*

$$F(\hat{\mathbf{x}}^N) - F(\mathbf{x}^*) \le \tfrac{2}{N(N+3)}\left[\tfrac{\mu}{\mathcal{C}}\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \tfrac{2\|\mathbf{L}\|^2\mathcal{C}}{\mu}\|\mathbf{y}^0\|^2 + \tfrac{2\mu\tilde{D}}{\mathcal{C}}\right], \tag{4.31}$$

*and*

$$\|\mathbf{L}\hat{\mathbf{x}}^N\| \le \tfrac{8\|\mathbf{L}\|}{N(N+3)}\left[3\sqrt{2\tilde{D} + \mathbf{V}(\mathbf{x}^0, \mathbf{x}^*)} + \tfrac{7\|\mathbf{L}\|\mathcal{C}}{\mu}\|\mathbf{y}^* - \mathbf{y}^0\|\right], \tag{4.32}$$

*where $\hat{\mathbf{x}}^N = \frac{2}{N(N+3)}\sum_{k=1}^N (k+1)\hat{\mathbf{x}}^k$.*

*Proof* It is easy to check that (4.30) satisfies conditions (3.10)-(3.14) and (4.26). Moreover, we have

$$\begin{aligned}\textstyle\sum_{k=1}^N\sum_{t=1}^{T_k}\frac{2mM^2\theta_k}{T_k(T_k+1)}\frac{t}{(t+1)\mu/\mathcal{C}+(t-1)\eta_k} &= \textstyle\sum_{k=1}^N\frac{2mM^2\theta_k\mathcal{C}}{T_k(T_k+1)\mu}\sum_{t=1}^{T_k}\frac{2t}{2(t+1)+(t-1)k}\\ &\le \textstyle\sum_{k=1}^N\frac{2mM^2\theta_k\mathcal{C}}{T_k(T_k+1)\mu}\left(\frac{1}{2}+\sum_{t=2}^{T_k}\frac{2t}{(t-1)(k+1)}\right)\\ &\le \textstyle\sum_{k=1}^N\frac{mM^2\mathcal{C}(k+1)}{T_k(T_k+1)\mu}+\sum_{k=1}^N\frac{8mM^2\mathcal{C}(T_k-1)}{T_k(T_k+1)\mu}\le\frac{2\mu\tilde{D}}{\mathcal{C}}.\end{aligned}$$

Therefore, by plugging in these values to (4.28), we have

$$Q(\hat{\mathbf{z}}^N;\mathbf{x}^*,\mathbf{y})\le\tfrac{2}{N(N+3)}\left[\tfrac{\mu}{\mathcal{C}}\mathbf{V}(\mathbf{x}^0,\mathbf{x}^*)+\tfrac{2\|\mathbf{L}\|^2\mathcal{C}}{\mu}\|\mathbf{y}^0\|^2+\tfrac{2\mu\tilde{D}}{\mathcal{C}}\right]+\tfrac{2}{N(N+3)}\langle\hat{\mathbf{s}},\mathbf{y}\rangle. \tag{4.33}$$

Furthermore, from (4.29), we have for $N\ge 2$

$$\begin{aligned}\|\hat{\mathbf{x}}^N-\mathbf{x}^{N-1}\|^2&\le\tfrac{8\mathcal{C}}{\mu(N+1)(N-1)}\left[\tfrac{\mu}{\mathcal{C}}\mathbf{V}(\mathbf{x}^0,\mathbf{x}^*)+\tfrac{2\|\mathbf{L}\|^2\mathcal{C}}{\mu}\|\mathbf{y}^0-\mathbf{y}^*\|^2+\tfrac{2\mu\tilde{D}}{\mathcal{C}}\right],\\ \|\mathbf{y}^*-\mathbf{y}^N\|^2&\le\tfrac{N\mu}{(N-1)\|\mathbf{L}\|^2\mathcal{C}}\left[\tfrac{\mu}{\mathcal{C}}\mathbf{V}(\mathbf{x}^0,\mathbf{x}^*)+\tfrac{2\|\mathbf{L}\|^2\mathcal{C}}{\mu}\|\mathbf{y}^0-\mathbf{y}^*\|^2+\tfrac{2\mu\tilde{D}}{\mathcal{C}}\right].\end{aligned} \tag{4.34}$$

Let $\mathbf{s}^N:=\frac{2}{N(N+3)}\hat{\mathbf{s}}$, then by using (4.34), we have for $N\ge 2$

$$\begin{aligned}\|\mathbf{s}^N\|&\le\tfrac{2}{N(N+3)}\left[(N+1)\|\mathbf{L}\|\|\hat{\mathbf{x}}^N-\mathbf{x}^{N-1}\|+\tfrac{4\|\mathbf{L}\|^2\mathcal{C}}{\mu}\|\mathbf{y}^N-\mathbf{y}^*\|+\tfrac{4\|\mathbf{L}\|^2\mathcal{C}}{\mu}\|\mathbf{y}^*-\mathbf{y}^0\|\right]\\ &\le\tfrac{8\|\mathbf{L}\|}{N(N+3)}\left[3\sqrt{2\tilde{D}+\mathbf{V}(\mathbf{x}^0,\mathbf{x}^*)+\tfrac{2\|\mathbf{L}\|^2\mathcal{C}^2}{\mu^2}\|\mathbf{y}^0-\mathbf{y}^*\|^2}+\tfrac{\|\mathbf{L}\|\mathcal{C}}{\mu}\|\mathbf{y}^*-\mathbf{y}^0\|\right]\\ &\le\tfrac{8\|\mathbf{L}\|}{N(N+3)}\left[3\sqrt{2\tilde{D}+\mathbf{V}(\mathbf{x}^0,\mathbf{x}^*)}+\tfrac{7\|\mathbf{L}\|\mathcal{C}}{\mu}\|\mathbf{y}^*-\mathbf{y}^0\|\right].\end{aligned}$$

From (4.33), we further have

$$g(\hat{\mathbf{s}}^N,\hat{\mathbf{z}}^N)\le\tfrac{2}{N(N+3)}\left[\tfrac{\mu}{\mathcal{C}}\mathbf{V}(\mathbf{x}^0,\mathbf{x}^*)+\tfrac{2\|\mathbf{L}\|^2\mathcal{C}}{\mu}\|\mathbf{y}^0\|^2+\tfrac{2\mu\tilde{D}}{\mathcal{C}}\right].$$

Applying Proposition 1 to the above two inequalities, the results in (4.31) and (4.32) follow immediately.

We now make some remarks about the results obtained in Theorem 4. Firstly, similar to the general convex case, the best choice for $\tilde{D}$ (cf. (4.30)) would be $\mathbf{V}(\mathbf{x}^0,\mathbf{x}^*)$ so that the first and the third terms in (4.33) are about the same order. If there exists an estimate $\mathcal{D}_{X^m}>0$ satisfying (4.18), we can set $\tilde{D}=\mathcal{D}^2_{X^m}$.

Secondly, the complexity of the DCS method for solving strongly convex problems follows from (4.31) and (4.32). For simplicity, let us assume that $X$ is bounded, $\tilde{D}=\mathcal{D}^2_{X^m}$ and $\mathbf{y}^0=\mathbf{0}$. We can see that the total number of inter-node communication rounds and intra-node subgradient evaluations performed by each agent for finding an $(\epsilon,\delta)$-solution of (1.7) can be bounded by

$$\mathcal{O}\left\{\max\left(\sqrt{\tfrac{\mu\mathcal{D}^2_{X^m}}{\mathcal{C}\epsilon}},\sqrt{\tfrac{\|\mathbf{L}\|}{\delta}\left(\mathcal{D}_{X^m}+\tfrac{\mathcal{C}\|\mathbf{L}\|\|\mathbf{y}^*\|}{\mu}\right)}\right)\right\}\quad\text{and}\quad\mathcal{O}\left\{\tfrac{mM^2\mathcal{C}}{\mu}\max\left(\tfrac{1}{\epsilon},\tfrac{\|\mathbf{L}\|\mathcal{C}}{\mu\delta}\left(\tfrac{1}{\mathcal{D}_{X^m}}+\tfrac{\mathcal{C}\|\mathbf{L}\|\|\mathbf{y}^*\|}{\mathcal{D}^2_{X^m}\mu}\right)\right)\right\}, \tag{4.35}$$

respectively. In particular, if $\epsilon$ and $\delta$ satisfy

$$\tfrac{\epsilon}{\delta}\le\tfrac{\mu^2\mathcal{D}^2_{X^m}}{\|\mathbf{L}\|\mathcal{C}(\mu\mathcal{D}_{X^m}+\mathcal{C}\|\mathbf{L}\|\|\mathbf{y}^*\|)}, \tag{4.36}$$

then the complexity bounds in (4.35), respectively, reduce to

$$\mathcal{O}\left\{\sqrt{\tfrac{\mu\mathcal{D}^2_{X^m}}{\mathcal{C}\epsilon}}\right\}\quad\text{and}\quad\mathcal{O}\left\{\tfrac{mM^2\mathcal{C}}{\mu\epsilon}\right\}. \tag{4.37}$$

Thirdly, we compare DCS method with the centralized mirror descent method [42] applied to (1.1). In the worst case, the Lipschitz constant and strongly convex modulus of $f$ in (1.1) can be bounded by $M_f\le mM$, and $\mu_f\ge m\mu$, respectively, and each iteration of the method will incur $m$ subgradient evaluations. Therefore, the total number of

subgradient evaluations performed by the mirror descent method for finding an $\epsilon$-solution of (1.1), i.e., a point $\bar{x} \in X$ such that $f(\bar{x}) - f^* \leq \epsilon$, can be bounded by

$$\mathcal{O}\left\{\frac{m^2 M^2 \mathcal{C}}{\mu\epsilon}\right\}. \tag{4.38}$$

Observed that the second bound in (4.37) states only the number of subgradient evaluations for each agent in the DCS method, we conclude that the total number of subgradient evaluations performed by DCS is comparable to the classic mirror descent method as long as (4.36) holds and hence not improvable in general for the nonsmooth strongly convex case.

## 5 Stochastic Decentralized Communication Sliding

In this section, we consider the stochastic case where only the noisy subgradient information of the functions $f_i$, $i = 1, \ldots, m$, is available or easier to compute. This situation happens when the function $f_i$'s are given either in the form of expectation or as the summation of lots of components. This setting has attracted considerable interest in recent decades for its applications in a broad spectrum of disciplines including machine learning, signal processing, and operations research. We present a stochastic communication sliding method, namely the stochastic decentralized communication sliding (SDCS) method, and show that the similar complexity bounds as in Section 4 can still be obtained in expectation or with high probability.

### 5.1 The SDCS Algorithm

The first-order information of the function $f_i$, $i = 1, \ldots, m$, can be accessed by a stochastic oracle (SO), which, given a point $u^t \in X$, outputs a vector $G_i(u^t, \xi_i^t)$ such that

$$\mathbb{E}[G_i(u^t, \xi_i^t)] = f_i'(u^t) \in \partial f_i(u^t), \tag{5.1}$$

$$\mathbb{E}[\|G_i(u^t, \xi_i^t) - f_i'(u^t)\|_*^2] \leq \sigma^2, \tag{5.2}$$

where $\xi_i^t$ is a random vector which models a source of uncertainty and is independent of the search point $u^t$, and the distribution $\mathbb{P}(\xi_i)$ is not known in advance. We call $G_i(u^t, \xi_i^t)$ a *stochastic subgradient* of $f_i$ at $u^t$.

The SDCS method can be obtained by simply replacing the exact subgradients in the CS procedure of Algorithm 3 with the stochastic subgradients obtained from SO. This difference is described in Algorithm 4.

**Algorithm 4** SDCS

The projection step (4.6)-(4.7) in the CS procedure of Algorithm 3 is replaced by

$$h^{t-1} = H(u^{t-1}, \xi^{t-1}), \tag{5.3}$$

$$u^t = \operatorname{argmin}_{u \in U}\left[\langle w + h^{t-1}, u\rangle + \eta V(x, u) + \eta\beta_t V(u^{t-1}, u)\right], \tag{5.4}$$

where $H(u^{t-1}, \xi^{t-1})$ is a stochastic subgradient of $\phi$ at $u^{t-1}$.

We add a few remarks about the SDCS algorithm. Firstly, as in DCS, no additional communications of the dual variables are required when the subgradient projection (5.4) is performed for $T_k$ times in the inner loop. This is because the same $w_i^k$ has been used throughout the $T_k$ iterations of the Stochastic CS procedure. Secondly, the problem will reduce to the deterministic case if there is no stochastic noise associated with the SO, i.e., when $\sigma = 0$ in (5.2). Therefore, in Section 6, we investigate the convergence analysis for the stochastic case first and then simplify the analysis for the deterministic case by setting $\sigma = 0$.

5.2 Convergence of SDCS on General Convex Functions

We now establish the main convergence properties of the SDCS algorithm. More specifically, we provide in Lemma 4 an estimate on the gap function defined in (2.7) together with stepsize policies which work for the general convex case with $\mu = 0$ (cf. (1.2)). The proof of this lemma can be found in Section 6.

**Lemma 4** *Let the iterates $(\hat{\mathbf{x}}^k, \mathbf{y}^k)$ for $k = 1, \ldots, N$ be generated by Algorithm 4 and $\hat{\mathbf{z}}^N$ be defined as $\hat{\mathbf{z}}^N := \left(\sum_{k=1}^N \theta_k\right)^{-1} \sum_{k=1}^N \theta_k(\hat{\mathbf{x}}^k, \mathbf{y}^k)$. Assume the objective $f_i$, $i = 1, \ldots, m$, are general nonsmooth convex functions, i.e., $\mu = 0$ and $M > 0$. Let the parameters $\{\alpha_k\}$, $\{\theta_k\}$, $\{\eta_k\}$, $\{\tau_k\}$ and $\{T_k\}$ in Algorithm 4 satisfy (3.10)-(3.14) and (4.10). Let the parameters $\{\lambda_t\}$ and $\{\beta_t\}$ in the CS procedure of Algorithm 4 be set as (4.11). Then, for all $\mathbf{z} \in X^m \times \mathbb{R}^{md}$,*

$$\begin{aligned} Q(\hat{\mathbf{z}}^N; \mathbf{z}) \le & \left(\sum_{k=1}^N \theta_k\right)^{-1} \left\{ \frac{(T_1+1)(T_1+2)\theta_1\eta_1}{T_1(T_1+3)} \mathbf{V}(\mathbf{x}^0, \mathbf{x}) + \frac{\theta_1\tau_1}{2}\|\mathbf{y}^0\|^2 + \langle \hat{\mathbf{s}}, \mathbf{y}\rangle \right. \\ & \left. + \sum_{k=1}^N \sum_{t=1}^{T_k} \sum_{i=1}^m \frac{2\theta_k}{T_k(T_k+3)} \left[ (t+1)\langle \delta_i^{t-1,k}, x_i - u_i^{t-1}\rangle + \frac{4(M^2 + \|\delta_i^{t-1,k}\|_*^2)}{\eta_k} \right] \right\}, \end{aligned} \tag{5.5}$$

*where $\hat{\mathbf{s}} := \theta_N \mathbf{L}(\hat{\mathbf{x}}^N - \mathbf{x}^{N-1}) + \theta_1\tau_1(\mathbf{y}^N - \mathbf{y}^0)$ and $Q$ is defined in (2.7). Furthermore, for any saddle point $(\mathbf{x}^*, \mathbf{y}^*)$ of (1.8), we have*

$$\begin{aligned} & \frac{\theta_N}{2}\left(1 - \frac{\|\mathbf{L}\|^2}{\eta_N\tau_N}\right) \max\{\eta_N\|\hat{\mathbf{x}}^N - \mathbf{x}^{N-1}\|^2, \tau_N\|\mathbf{y}^* - \mathbf{y}^N\|^2\} \\ & \le \frac{(T_1+1)(T_1+2)\theta_1\eta_1}{T_1(T_1+3)} \mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \frac{\theta_1\tau_1}{2}\|\mathbf{y}^* - \mathbf{y}^0\|^2 \\ & \quad + \sum_{k=1}^N \sum_{t=1}^{T_k} \sum_{i=1}^m \frac{2\theta_k}{T_k(T_k+3)} \left[ (t+1)\langle \delta_i^{t-1,k}, x_i^* - u_i^{t-1}\rangle + \frac{4(M^2 + \|\delta_i^{t-1,k}\|_*^2)}{\eta_k} \right]. \end{aligned} \tag{5.6}$$

In the following theorem, we provide a specific selection of $\{\alpha_k\}$, $\{\theta_k\}$, $\{\eta_k\}$, $\{\tau_k\}$ and $\{T_k\}$ satisfying (3.10)-(3.14) and (4.10). Also, by using Lemma 4 and Proposition 1, we establish the complexity of the SDCS method for computing an $(\epsilon, \delta)$-solution of problem (1.7) in expectation when the objective functions are general convex.

**Theorem 5** *Let $\mathbf{x}^*$ be an optimal solution of (1.7), the parameters $\{\lambda_t\}$ and $\{\beta_t\}$ in the CS procedure of Algorithm 4 be set as (4.11), and suppose that $\{\alpha_k\}$, $\{\theta_k\}$, $\{\eta_k\}$, $\{\tau_k\}$ and $\{T_k\}$ are set to*

$$\alpha_k = \theta_k = 1, \ \eta_k = 2\|\mathbf{L}\|, \ \tau_k = \|\mathbf{L}\|, \ \text{and } T_k = \left\lceil \frac{m(M^2+\sigma^2)N}{\|\mathbf{L}\|^2\tilde{D}} \right\rceil, \quad \forall k = 1, \ldots, N, \tag{5.7}$$

*for some $\tilde{D} > 0$. Then, under Assumptions (5.1) and (5.2), we have for any $N \ge 1$*

$$\mathbb{E}[F(\hat{\mathbf{x}}^k) - F(\mathbf{x}^*)] \le \frac{\|\mathbf{L}\|}{N}\left[3\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \tfrac{1}{2}\|\mathbf{y}^0\|^2 + 4\tilde{D}\right], \tag{5.8}$$

*and*

$$\mathbb{E}[\|\mathbf{L}\hat{\mathbf{x}}^N\|] \le \frac{\|\mathbf{L}\|}{N}\left[3\sqrt{6\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + 8\tilde{D}} + 4\|\mathbf{y}^* - \mathbf{y}^0\|\right]. \tag{5.9}$$

*where $\hat{\mathbf{x}}^N = \frac{1}{N}\sum_{k=1}^N \hat{\mathbf{x}}^k$.*

*Proof* It is easy to check that (5.7) satisfies conditions (3.10)-(3.14) and (4.10). Moreover, by (2.9), we can obtain

$$\begin{aligned} g(\hat{\mathbf{s}}^N, \hat{\mathbf{z}}^N) = & \max_{\mathbf{y}} Q(\hat{\mathbf{z}}^N; \mathbf{x}^*, \mathbf{y}) - \left(\sum_{k=1}^N \theta_k\right)^{-1} \langle \hat{\mathbf{s}}, \mathbf{y}\rangle \\ \le & \left(\sum_{k=1}^N \theta_k\right)^{-1} \left\{ \frac{(T_1+1)(T_1+2)\theta_1\eta_1}{T_1(T_1+3)} \mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \frac{\theta_1\tau_1}{2}\|\mathbf{y}^0\|^2 \right. \\ & \left. + \sum_{k=1}^N \sum_{t=1}^{T_k} \sum_{i=1}^m \frac{2\theta_k}{T_k(T_k+3)} \left[ (t+1)\langle \delta_i^{t-1,k}, x_i^* - u_i^{t-1}\rangle + \frac{4(M^2 + \|\delta_i^{t-1,k}\|_*^2)}{\eta_k} \right] \right\}, \end{aligned} \tag{5.10}$$

where $\mathbf{s}^N = \left(\sum_{k=1}^N \theta_k\right)^{-1} \hat{\mathbf{s}}$. Particularly, from Assumption (5.1) and (5.2),

$$\mathbb{E}[\delta_i^{t-1,k}] = 0, \quad \mathbb{E}[\|\delta_i^{t-1,k}\|_*^2] \leq \sigma^2, \ \forall i \in \{1, \ldots, m\}, \ t \geq 1, \ k \geq 1,$$

and from (5.7)

$$\tfrac{(T_1+1)(T_1+2)}{T_1(T_1+3)} = 1 + \tfrac{2}{T_1^2+3T_1} \leq \tfrac{3}{2}.$$

Therefore, by taking expectation over both sides of (5.10) and plugging in these values into (5.10), we have

$$\begin{aligned} \mathbb{E}[g(\hat{\mathbf{s}}^N, \hat{\mathbf{z}}^N)] &\leq \left(\textstyle\sum_{k=1}^N \theta_k\right)^{-1} \left\{ \tfrac{(T_1+1)(T_1+2)\theta_1\eta_1}{T_1(T_1+3)} \mathbf{V}(\mathbf{x}^0, \mathbf{x}) + \tfrac{\theta_1\tau_1}{2}\|\mathbf{y}^0\|^2 + \textstyle\sum_{k=1}^N \tfrac{8m(M^2+\sigma^2)\theta_k}{(T_k+3)\eta_k} \right\} \\ &\leq \tfrac{\|\mathbf{L}\|}{N}\left[3\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \tfrac{1}{2}\|\mathbf{y}^0\|^2 + 4\tilde{D}\right], \end{aligned} \tag{5.11}$$

with

$$\mathbb{E}[\|\hat{\mathbf{s}}^N\|] = \tfrac{1}{N}\mathbb{E}[\|\hat{\mathbf{s}}\|] \leq \tfrac{\|\mathbf{L}\|}{N}\mathbb{E}\left[\|\hat{\mathbf{x}}^N - \mathbf{x}^{N-1}\| + \|\mathbf{y}^N - \mathbf{y}^*\| + \|\mathbf{y}^* - \mathbf{y}^0\|\right].$$

Note that from (5.6) and Jensen's inequality, we have

$$\begin{aligned} (\mathbb{E}[\|\hat{\mathbf{x}}^N - \mathbf{x}^{N-1}\|])^2 \leq \mathbb{E}[\|\hat{\mathbf{x}}^N - \mathbf{x}^{N-1}\|^2] \leq 6\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \|\mathbf{y}^* - \mathbf{y}^0\| + 8\tilde{D}, \\ (\mathbb{E}[\|\mathbf{y}^* - \mathbf{y}^N\|])^2 \leq \mathbb{E}[\|\mathbf{y}^* - \mathbf{y}^N\|^2] \leq 12\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + 2\|\mathbf{y}^* - \mathbf{y}^0\| + 16\tilde{D}. \end{aligned}$$

Hence,

$$\mathbb{E}[\|\hat{\mathbf{s}}^N\|] \leq \tfrac{\|\mathbf{L}\|}{N}\left[3\sqrt{6\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + 8\tilde{D}} + 4\|\mathbf{y}^* - \mathbf{y}^0\|\right].$$

Applying Proposition 1 to the above inequality and (5.11), the results in (5.8) and (5.9) follow immediately.

We now make some observations about the results obtained in Theorem 5. Firstly, one can choose any $\tilde{D} > 0$ (e.g., $\tilde{D} = 1$) in (5.7), however, the best selection of $\tilde{D}$ would be $\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*)$ so that the first and third terms in (5.11) are about the same order. In practice, if there exists an estimate $\mathcal{D}_{X^m} > 0$ satisfying (4.18), we can set $\tilde{D} = \mathcal{D}_{X^m}^2$.

Secondly, the complexity of SDCS method immediately follows from (5.8) and (5.9). Under the above assumption, with $\tilde{D} = \mathcal{D}_{X^m}^2$ and $\mathbf{y}^0 = \mathbf{0}$, we can see that the total number of inter-node communication rounds and intra-node subgradient evaluations required by each agent for finding a stochastic $(\epsilon, \delta)$-solution of (1.7) can be bounded by

$$\mathcal{O}\left\{\|\mathbf{L}\| \max\left(\tfrac{\mathcal{D}_{X^m}^2}{\epsilon}, \tfrac{\mathcal{D}_{X^m}+\|\mathbf{y}^*\|}{\delta}\right)\right\} \quad \text{and} \quad \mathcal{O}\left\{m(M^2+\sigma^2)\max\left(\tfrac{\mathcal{D}_{X^m}^2}{\epsilon^2}, \tfrac{\mathcal{D}_{X^m}^2+\|\mathbf{y}^*\|^2}{\mathcal{D}_{X^m}^2\delta^2}\right)\right\}, \tag{5.12}$$

respectively. In particular, if $\epsilon$ and $\delta$ satisfy (4.20), the above complexity bounds, respectively, reduce to

$$\mathcal{O}\left\{\tfrac{\|\mathbf{L}\|\mathcal{D}_{X^m}^2}{\epsilon}\right\} \quad \text{and} \quad \mathcal{O}\left\{\tfrac{m(M^2+\sigma^2)\mathcal{D}_{X^m}^2}{\epsilon^2}\right\}. \tag{5.13}$$

In particular, we can show that the total number stochastic subgradients that SDCS requires is comparable to the mirror-descent stochastic approximation in [41]. This implies that the sample complexity for decentralized stochastic optimization are still optimal (as the centralized one), even after we skip many communication rounds.

5.3 Convergence of SDCS on Strongly Convex Functions

We now provide in Lemma 5 an estimate on the gap function defined in (2.7) together with stepsize policies which work for the strongly convex case with $\mu > 0$ (cf. (1.2)). The proof of this lemma can be found in Section 6.

Note that throughout this subsection, we assume that the prox-functions $V_i(\cdot,\cdot)$, $i = 1,\ldots,m$, (cf. (2.2)) are growing quadratically with the quadratic growth constant $\mathcal{C}$, i.e., (4.25) holds.

**Lemma 5** *Let the iterates $(\hat{\mathbf{x}}^k, \mathbf{y}^k)$, $k = 1,\ldots,N$ be generated by Algorithm 4 and $\hat{\mathbf{z}}^N$ be defined as $\hat{\mathbf{z}}^N := \left(\sum_{k=1}^N \theta_k\right)^{-1} \sum_{k=1}^N \theta_k(\hat{\mathbf{x}}^k, \mathbf{y}^k)$. Assume the objective $f_i$, $i = 1,\ldots,m$ are strongly convex functions, i.e., $\mu, M > 0$. Let the parameters $\{\alpha_k\}$, $\{\theta_k\}$, $\{\eta_k\}$ and $\{\tau_k\}$ in Algorithm 4 satisfy* (3.10)-(3.14) *and* (4.26). *Let the parameters $\{\lambda_t\}$ and $\{\beta_t\}$ in the CS procedure of Algorithm 4 be set as* (4.27). *Then, for all $\mathbf{z} \in X^m \times \mathbb{R}^{md}$,*

$$\begin{aligned} Q(\hat{\mathbf{z}}^N; \mathbf{z}) \;\le\; & \left(\textstyle\sum_{k=1}^N \theta_k\right)^{-1} \left\{ \theta_1\eta_1 \mathbf{V}(\mathbf{x}^0, \mathbf{x}) + \tfrac{\theta_1\tau_1}{2}\|\mathbf{y}^0\|^2 + \langle \hat{\mathbf{s}}, \mathbf{y}\rangle \right. \\ & \left. + \textstyle\sum_{k=1}^N \sum_{t=1}^{T_k} \sum_{i=1}^m \frac{2\theta_k}{T_k(T_k+1)} \left[ t\langle \delta_i^{t-1,k}, x_i - u_i^{t-1}\rangle + \frac{2t(M^2 + \|\delta_i^{t-1,k}\|_*^2)}{(t+1)\mu/\mathcal{C} + (t-1)\eta_k} \right] \right\}, \end{aligned} \tag{5.14}$$

*where $\hat{\mathbf{s}} := \theta_N \mathbf{L}(\hat{\mathbf{x}}^N - \mathbf{x}^{N-1}) + \theta_1\tau_1(\mathbf{y}^N - \mathbf{y}^0)$ and $Q$ is defined in* (2.7). *Furthermore, for any saddle point $(\mathbf{x}^*, \mathbf{y}^*)$ of* (1.8), *we have*

$$\begin{aligned} & \tfrac{\theta_N}{2}\left(1 - \tfrac{\|\mathbf{L}\|^2}{\eta_N\tau_N}\right) \max\{\eta_N\|\hat{\mathbf{x}}^N - \mathbf{x}^{N-1}\|^2, \tau_N\|\mathbf{y}^* - \mathbf{y}^N\|^2\} \\ & \quad \le \theta_1\eta_1 \mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \tfrac{\theta_1\tau_1}{2}\|\mathbf{y}^* - \mathbf{y}^0\|^2 \\ & \qquad + \textstyle\sum_{k=1}^N \sum_{t=1}^{T_k} \sum_{i=1}^m \frac{2\theta_k}{T_k(T_k+1)} \left[ t\langle \delta_i^{t-1,k}, x_i^* - u_i^{t-1}\rangle + \frac{2t(M^2 + \|\delta_i^{t-1,k}\|_*^2)}{(t+1)\mu/\mathcal{C} + (t-1)\eta_k} \right]. \end{aligned} \tag{5.15}$$

In the following theorem, we provide a specific selection of $\{\alpha_k\}$, $\{\theta_k\}$, $\{\eta_k\}$, $\{\tau_k\}$ and $\{T_k\}$ satisfying (3.10)-(3.14) and (4.10). Also, by using Lemma 5 and Proposition 1, we establish the complexity of the SDCS method for computing an $(\epsilon, \delta)$-solution of problem (1.7) in expectation when the objective functions are strongly convex. Similar to the deterministic case, we choose variable stepsizes rather than constant stepsizes.

**Theorem 6** *Let $\mathbf{x}^*$ be an optimal solution of* (1.7), *the parameters $\{\lambda_t\}$ and $\{\beta_t\}$ in the CS procedure of Algorithm 4 be set as* (4.27), *and suppose that $\{\alpha_k\}$, $\{\theta_k\}$, $\{\eta_k\}$, $\{\tau_k\}$ and $\{T_k\}$ are set to*

$$\begin{aligned} & \alpha_k = \tfrac{k}{k+1},\ \theta_k = k+1,\ \eta_k = \tfrac{k\mu}{2\mathcal{C}},\ \tau_k = \tfrac{4\|\mathbf{L}\|^2\mathcal{C}}{(k+1)\mu},\ \text{and} \\ & T_k = \left\lceil \sqrt{\tfrac{m(M^2+\sigma^2)}{\tilde{D}}}\, \tfrac{2N\mathcal{C}}{\mu} \max\left\{ \sqrt{\tfrac{m(M^2+\sigma^2)}{\tilde{D}}}\, \tfrac{8\mathcal{C}}{\mu}, 1 \right\} \right\rceil, \quad \forall k = 1,\ldots,N, \end{aligned} \tag{5.16}$$

*for some $\tilde{D} > 0$. Then, under Assumptions* (5.1) *and* (5.2), *we have for any $N \ge 2$*

$$\mathbb{E}[F(\bar{\mathbf{x}}^N) - F(\mathbf{x}^*) \le \tfrac{2}{N(N+3)} \left[ \tfrac{\mu}{\mathcal{C}} \mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \tfrac{2\|\mathbf{L}\|^2\mathcal{C}}{\mu}\|\mathbf{y}^0\|^2 + \tfrac{2\mu\tilde{D}}{\mathcal{C}} \right], \tag{5.17}$$

*and*

$$\mathbb{E}[\|\mathbf{L}\hat{\mathbf{x}}^N\|] \le \tfrac{8\|\mathbf{L}\|}{N(N+3)} \left[ 3\sqrt{2\tilde{D} + \mathbf{V}(\mathbf{x}^0, \mathbf{x}^*)} + \tfrac{7\|\mathbf{L}\|\mathcal{C}}{\mu}\|\mathbf{y}^* - \mathbf{y}^0\| \right], \tag{5.18}$$

*where $\hat{\mathbf{x}}^N = \frac{2}{N(N+3)} \sum_{k=1}^N (k+1)\hat{\mathbf{x}}^k$.*

*Proof* It is easy to check that (5.16) satisfies conditions (3.10)-(3.14) and (4.26). Similarly, by (2.9), Assumption (5.1) and (5.2), we can obtain

$$\mathbb{E}[g(\hat{\mathbf{s}}^N, \hat{\mathbf{z}}^N)] \le \left(\textstyle\sum_{k=1}^N \theta_k\right)^{-1} \left\{ \theta_1\eta_1 \mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \tfrac{\theta_1\tau_1}{2}\|\mathbf{y}^0\|^2 + \textstyle\sum_{k=1}^N \sum_{t=1}^{T_k} \sum_{i=1}^m \frac{2\theta_k}{T_k(T_k+1)} \left[ \frac{2t(M^2+\sigma^2)}{(t+1)\mu/\mathcal{C} + (t-1)\eta_k} \right] \right\}, \tag{5.19}$$

where $\mathbf{s}^N = \left(\sum_{k=1}^N \theta_k\right)^{-1} \hat{\mathbf{s}}$. Particularly, from (5.16), we have

$$\begin{aligned}\sum_{k=1}^N \sum_{t=1}^{T_k} \frac{4m(M^2+\sigma^2)\theta_k}{T_k(T_k+1)} \frac{t}{(t+1)\mu/\mathcal{C}+(t-1)\eta_k} &= \sum_{k=1}^N \frac{4m(M^2+\sigma^2)\mathcal{C}\theta_k}{T_k(T_k+1)\mu} \sum_{t=1}^{T_k} \frac{2t}{2(t+1)+(t-1)k} \\ &\le \sum_{k=1}^N \frac{4m(M^2+\sigma^2)\mathcal{C}\theta_k}{T_k(T_k+1)\mu} \left(\frac{1}{2} + \sum_{t=2}^{T_k} \frac{2t}{(t-1)(k+1)}\right) \\ &\le \sum_{k=1}^N \frac{2m(M^2+\sigma^2)\mathcal{C}(k+1)}{T_k(T_k+1)\mu} + \sum_{k=1}^N \frac{16m(M^2+\sigma^2)\mathcal{C}(T_k-1)}{T_k(T_k+1)\mu} \le \frac{2\mu\tilde{D}}{\mathcal{C}}.\end{aligned}$$

Therefore, by plugging in these values into (5.19), we have

$$\mathbb{E}[g(\hat{\mathbf{s}}^N, \hat{\mathbf{z}}^N)] \le \frac{2}{N(N+3)} \left[\frac{\mu}{\mathcal{C}} \mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \frac{2\|L\|^2\mathcal{C}}{\mu}\|\mathbf{y}^0\|^2 + \frac{2\mu\tilde{D}}{\mathcal{C}}\right], \tag{5.20}$$

with

$$\mathbb{E}[\|\hat{\mathbf{s}}^N\|] = \frac{2}{N(N+3)}\mathbb{E}[\|\hat{\mathbf{s}}\|] \le \frac{2\|\mathbf{L}\|}{N(N+3)}\mathbb{E}\left[(N+1)\|\hat{\mathbf{x}}^N - \mathbf{x}^{N-1}\| + \frac{4\|\mathbf{L}\|\mathcal{C}}{\mu}(\|\mathbf{y}^N - \mathbf{y}^*\| + \|\mathbf{y}^* - \mathbf{y}^0\|)\right].$$

Note that from (5.15), we have, for any $N \ge 2$,

$$\begin{aligned}\mathbb{E}[\|\hat{\mathbf{x}}^N - \mathbf{x}^{N-1}\|^2] &\le \frac{8}{(N+1)(N-1)} \left[\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \frac{2\|\mathbf{L}\|^2\mathcal{C}^2}{\mu^2}\|\mathbf{y}^0 - \mathbf{y}^*\|^2 + 2\tilde{D}\right], \\ \mathbb{E}[\|\mathbf{y}^* - \mathbf{y}^N\|^2] &\le \frac{N\mu}{(N-1)\|\mathbf{L}\|^2\mathcal{C}} \left[\frac{\mu}{\mathcal{C}}\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \frac{2\|\mathbf{L}\|^2\mathcal{C}}{\mu}\|\mathbf{y}^0 - \mathbf{y}^*\|^2 + \frac{2\mu\tilde{D}}{\mathcal{C}}\right].\end{aligned}$$

Hence, in view of the above three relations and Jensen's inequality, we obtain

$$\begin{aligned}\mathbb{E}[\|\hat{\mathbf{s}}^N\|] &\le \frac{8\|\mathbf{L}\|}{N(N+3)} \left[3\sqrt{2\tilde{D} + \mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \frac{2\|\mathbf{L}\|^2\mathcal{C}^2}{\mu^2}\|\mathbf{y}^0 - \mathbf{y}^*\|^2} + \frac{\|\mathbf{L}\|\mathcal{C}}{\mu}\|\mathbf{y}^* - \mathbf{y}^0\|\right] \\ &\le \frac{8\|\mathbf{L}\|}{N(N+3)} \left[3\sqrt{2\tilde{D} + \mathbf{V}(\mathbf{x}^0, \mathbf{x}^*)} + \frac{7\|\mathbf{L}\|\mathcal{C}}{\mu}\|\mathbf{y}^* - \mathbf{y}^0\|\right].\end{aligned}$$

Applying Proposition 1 to the above inequality and (5.20), the results in (5.17) and (5.18) follow immediately.

We now make some observations about the results obtained in Theorem 6. Firstly, similar to the general convex case, the best choice for $\tilde{D}$ (cf. (5.16)) would be $\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*)$ so that the first and the third terms in (5.20) are about the same order. If there exists an estimate $\mathcal{D}_{X^m} > 0$ satisfying (4.18), we can set $\tilde{D} = \mathcal{D}^2_{X^m}$.

Secondly, the complexity of SDCS method for solving strongly convex problems follows from (5.17) and (5.18). Under the above assumption, with $\tilde{D} = \mathcal{D}^2_{X^m}$ and $\mathbf{y}^0 = \mathbf{0}$, the total number of inter-node communication rounds and intra-node subgradient evaluations performed by each agent for finding a stochastic $(\epsilon, \delta)$-solution of (1.7) can be bounded by

$$\mathcal{O}\left\{\max\left(\sqrt{\frac{\mu\mathcal{D}^2_{X^m}}{\mathcal{C}\epsilon}}, \sqrt{\frac{\|\mathbf{L}\|}{\delta}\left(\mathcal{D}_{X^m} + \frac{\mathcal{C}\|\mathbf{L}\|\|\mathbf{y}^*\|}{\mu}\right)}\right)\right\} \quad \text{and} \quad \mathcal{O}\left\{\frac{m(M^2+\sigma^2)\mathcal{C}}{\mu}\max\left(\frac{1}{\epsilon}, \frac{\mathcal{C}\|\mathbf{L}\|}{\mu\delta}\left(\frac{1}{\mathcal{D}_{X^m}} + \frac{\mathcal{C}\|\mathbf{L}\|\|\mathbf{y}^*\|}{\mathcal{D}^2_{X^m}\mu}\right)\right)\right\}, \tag{5.21}$$

respectively. In particular, if $\epsilon$ and $\delta$ satisfy (4.36), the above complexity bounds, respectively, reduce to

$$\mathcal{O}\left\{\sqrt{\frac{\mu\mathcal{D}^2_{X^m}}{\mathcal{C}\epsilon}}\right\} \quad \text{and} \quad \mathcal{O}\left\{\frac{m(M^2+\sigma^2)\mathcal{C}}{\mu\epsilon}\right\}. \tag{5.22}$$

We can see that the total number of stochastic subgradient computations is comparable to the optimal complexity bound obtained in [16,17] for stochastic strongly convex case in the centralized case.

5.4 High Probability Results

All of the results stated in Section 5.2-5.3 are established in terms of expectation. In order to provide high probability results for SDCS method, we additionally need the following "light-tail" assumption:

$$\mathbb{E}[\exp\{\|G_i(u^t,\xi_i^t)-f_i'(u^t)\|_*^2/\sigma^2\}]\le\exp\{1\}. \tag{5.23}$$

Note that (5.23) is stronger than (5.2), since it implies (5.2) by Jensen's inequality. Moreover, we also assume that there exists $\bar{\mathbf{V}}(\mathbf{x}^*)$ s.t.

$$\bar{\mathbf{V}}(\mathbf{x}^*):=\textstyle\sum_{i=1}^m\bar{V}_i(x_i^*):=\sum_{i=1}^m\max_{x_i\in X_i}V_i(x_i^*,x_i). \tag{5.24}$$

The following theorem provides a large deviation result for the gap function $g(\hat{\mathbf{s}}^N,\hat{\mathbf{z}}^N)$ when our objective functions $f_i$, $i=1,\ldots,m$ are general nonsmooth convex functions.

**Theorem 7** *Assume the objective $f_i$, $i=1,\ldots,m$ are general nonsmooth convex functions, i.e., $\mu=0$ and $M>0$. Let Assumptions* (5.1), (5.2) *and* (5.23) *hold, the parameters $\{\alpha_k\}$, $\{\theta_k\}$, $\{\eta_k\}$, $\{\tau_k\}$ and $\{T_k\}$ in Algorithm 4 satisfy* (3.10)-(3.14)*, and* (4.10)*, and the parameters $\{\lambda_t\}$ and $\{\beta_t\}$ in the CS procedure of Algorithm 4 be set as* (4.11)*. In addition, if $X_i$'s are compact, then for any $\zeta>0$ and $N\ge1$, we have*

$$\text{Prob}\left\{g(\hat{\mathbf{s}}^N,\hat{\mathbf{z}}^N)\ge\mathcal{B}_d(N)+\zeta\mathcal{B}_p(N)\right\}\le\exp\{-\zeta^2/3\}+\exp\{-\zeta\}, \tag{5.25}$$

*where*

$$\mathcal{B}_d(N):=\left(\textstyle\sum_{k=1}^N\theta_k\right)^{-1}\left[\frac{(T_1+1)(T_1+2)\theta_1\eta_1}{T_1(T_1+3)}\mathbf{V}(\mathbf{x}^0,\mathbf{x}^*)+\frac{\theta_1\tau_1}{2}\|\mathbf{y}^0\|^2+\sum_{k=1}^N\frac{8m(M^2+\sigma^2)\theta_k}{\eta_k(T_k+3)}\right], \tag{5.26}$$

*and*

$$\mathcal{B}_p(N):=\left(\textstyle\sum_{k=1}^N\theta_k\right)^{-1}\left\{\sigma\left[2\bar{\mathbf{V}}(\mathbf{x}^*)\sum_{k=1}^N\sum_{t=1}^{T_k}\left(\frac{\theta_k\lambda_t}{\sum_{t=1}^{T_k}\lambda_t}\right)^2\right]^{1/2}+\sum_{k=1}^N\sum_{t=1}^{T_k}\sum_{i=1}^m\frac{\sigma^2\theta_k\lambda_t}{\left(\sum_{t=1}^{T_k}\lambda_t\right)\eta_k\beta_t}\right\}. \tag{5.27}$$

In the next corollary, we establish the rate of convergence of SDCS in terms of both primal and feasibility (or consistency) residuals are of order $\mathcal{O}(1/N)$ with high probability when the objective functions are nonsmooth and convex.

**Corollary 1** *Let $\mathbf{x}^*$ be an optimal solution of* (1.7)*, the parameters $\{\lambda_t\}$ and $\{\beta_t\}$ in the CS procedure of Algorithm 4 be set as* (4.11)*, and suppose that $\{\alpha_k\}$, $\{\theta_k\}$, $\{\eta_k\}$, $\{\tau_k\}$ and $\{T_k\}$ are set to* (5.7) *with $\tilde{D}=\bar{\mathbf{V}}(\mathbf{x}^*)$. Under Assumptions* (5.1), (5.2) *and* (5.23)*, we have for any $N\ge1$ and $\zeta>0$*

$$\text{Prob}\left\{F(\hat{\mathbf{x}}^N)-F(\mathbf{x}^*)\ge\tfrac{\|\mathbf{L}\|}{N}\left[(7+8\zeta)\bar{\mathbf{V}}(\mathbf{x}^*)+\tfrac{1}{2}\|\mathbf{y}^0\|^2\right]\right\}\le\exp\{-\zeta^2/3\}+\exp\{-\zeta\}, \tag{5.28}$$

*and*

$$\text{Prob}\left\{\|\mathbf{L}\hat{\mathbf{x}}^N\|^2\ge\tfrac{18\|\mathbf{L}\|^2}{N^2}\left[(7+8\zeta)\bar{\mathbf{V}}(\mathbf{x}^*)+\tfrac{2}{3}\|\mathbf{y}^*-\mathbf{y}^0\|^2\right]\right\}\le\exp\{-\zeta^2/3\}+\exp\{-\zeta\}. \tag{5.29}$$

*Proof* Observe that by the definition of $\lambda_t$ in (4.11),

$$\begin{aligned}\textstyle\sum_{t=1}^{T_k}\left[\frac{\theta_k\lambda_t}{\sum_{t=1}^{T_k}\lambda_t}\right]^2&=\left(\tfrac{2}{T_k(T_k+3)}\right)^2\textstyle\sum_{t=1}^{T_k}(t+1)^2\\&=\left(\tfrac{2}{T_k(T_k+3)}\right)^2\tfrac{(T_k+1)(T_k+2)(2T_k+3)}{6}\le\tfrac{8}{3T_k},\end{aligned}$$

which together with (5.27) then imply that

$$\mathcal{B}_p(N)\le\tfrac{1}{N}\left\{\sigma\left[2\bar{\mathbf{V}}(\mathbf{x}^*)\textstyle\sum_{k=1}^N\frac{8}{3T_k}\right]^{1/2}+\sum_{k=1}^N\frac{8m\sigma^2}{\|\mathbf{L}\|(T_k+3)}\right\}$$

$$\leq \tfrac{4\|\mathbf{L}\|}{N}\left\{\sqrt{\tfrac{\bar{\mathbf{V}}(\mathbf{x}^*)\tilde{D}}{3m}}+\tilde{D}\right\} \leq \tfrac{8\|\mathbf{L}\|\bar{\mathbf{V}}(\mathbf{x}^*)}{N}.$$

Hence, (5.28) follows from the above relation, (5.25) and Proposition 1. Note that from (5.6) and plugging in (5.7) with $\tilde{D} = \bar{\mathbf{V}}(\mathbf{x}^*)$, we obtain

$$\begin{aligned}
\|\hat{\mathbf{s}}^N\|^2 &= \left(\textstyle\sum_{k=1}^N \theta_k\right)^{-2}\|\hat{\mathbf{s}}\|^2 \\
&\leq \left(\textstyle\sum_{k=1}^N \theta_k\right)^{-2}\left\{3\theta_N^2\|\mathbf{L}\|^2\|\hat{\mathbf{x}}^N-\mathbf{x}^{N-1}\|^2+3\theta_1^2\tau_1^2\left(\|\mathbf{y}^N-\mathbf{y}^*\|^2+\|\mathbf{y}^*-\mathbf{y}^0\|^2\right)\right\} \\
&\leq \tfrac{3\|\mathbf{L}\|^2}{N^2}\Bigg\{18\mathbf{V}(\mathbf{x}^0,\mathbf{x}^*)+4\|\mathbf{y}^*-\mathbf{y}^0\|^2 \\
&\qquad + \textstyle\sum_{k=1}^N\sum_{t=1}^{T_k}\sum_{i=1}^m \frac{12\theta_k}{T_k(T_k+3)\|\mathbf{L}\|}\left[(t+1)\langle\delta_i^{t-1,k},x_i^*-u_i^{t-1}\rangle+\tfrac{4(M^2+\|\delta_i^{t-1,k}\|_*^2)}{\eta_k}\right]\Bigg\}.
\end{aligned}$$

Hence, similarly, we have

$$\text{Prob}\left\{\|\hat{\mathbf{s}}^N\|^2 \geq \tfrac{18\|\mathbf{L}\|^2}{N^2}\left[(7+8\zeta)\bar{\mathbf{V}}(\mathbf{x}^*)+\tfrac{2}{3}\|\mathbf{y}^*-\mathbf{y}^0\|^2\right]\right\} \leq \exp\{-\zeta^2/3\}+\exp\{-\zeta\},$$

which in view of Proposition 1 immediately implies (5.29).

## 6 Convergence Analysis

This section is devoted to prove the main lemmas in Section 3, 4 and 5, which establish the convergence results of the decentralized primal-dual method, the deterministic and stochastic decentralized communication sliding methods, respectively. After introducing some general results about these algorithms, we provide the proofs for Lemma 1-5 and Theorem 7.

The following lemma below characterizes the solution of the primal and dual projection steps (3.2), (3.3) (also (3.6), (3.8)) as well as the projection in inner loop (4.7). The proof of this result can be found in Lemma 2 of [16].

**Lemma 6** *Let the convex function $q: U \to \mathbb{R}$, the points $\bar{x}, \bar{y} \in U$ and the scalars $\mu_1, \mu_2 \in \mathbb{R}$ be given. Let $\omega: U \to \mathbb{R}$ be a differentiable convex function and $V(x,z)$ be defined in* (2.2)*. If*

$$u^* \in \operatorname{argmin}\left\{q(u)+\mu_1 V(\bar{x},u)+\mu_2 V(\bar{y},u) : u \in U\right\},$$

*then for any $u \in U$, we have*

$$q(u^*)+\mu_1 V(\bar{x},u^*)+\mu_2 V(\bar{y},u^*) \leq q(u)+\mu_1 V(\bar{x},u)+\mu_2 V(\bar{y},u)-(\mu_1+\mu_2)V(u^*,u).$$

We are now ready to provide a proof for Lemma 1 which establishes the convergence property for the decentralized primal-dual method. Note that this result also builds up the basic recursion for the outer loop of the DCS and SDCS methods.

**Proof of Lemma 1:** Note that applying Lemma 6 to (3.6) and (3.8), we have

$$\langle v_i^k, y_i - y_i^k\rangle \leq \tfrac{\tau_k}{2}\left[\|y_i-y_i^{k-1}\|^2-\|y_i-y_i^k\|^2-\|y_i^{k-1}-y_i^k\|^2\right],\ \forall y_i \in \mathbb{R}^d,$$

$$\langle w_i^k, x_i^k - x_i\rangle + f_i(x_i^k)-f_i(x_i) \leq \eta_k\left[V_i(x_i^{k-1},x_i)-V_i(x_i^k,x_i)-V_i(x_i^{k-1},x_i^k)\right],\ \forall x_i \in X_i,$$

which in view of the definition of $Q$ and $\mathbf{V}(\cdot,\cdot)$ in (2.7) and (2.5), respectively, we can obtain

$$\begin{aligned}
Q(\mathbf{x}^k,\mathbf{y}^k;\mathbf{z}) &= F(\mathbf{x}^k)-F(\mathbf{x})+\langle\mathbf{L}\mathbf{x}^k,\mathbf{y}\rangle-\langle\mathbf{L}\mathbf{x},\mathbf{y}^k\rangle \\
&\leq \langle\mathbf{L}(\mathbf{x}^k-\tilde{\mathbf{x}}^k),\mathbf{y}-\mathbf{y}^k\rangle+\eta_k\left[\mathbf{V}(\mathbf{x}^{k-1},\mathbf{x})-\mathbf{V}(\mathbf{x}^k,\mathbf{x})-\mathbf{V}(\mathbf{x}^{k-1},\mathbf{x}^k)\right]
\end{aligned}$$

$$+\ \tfrac{\tau_k}{2}\left[\|\mathbf{y}-\mathbf{y}^{k-1}\|^2-\|\mathbf{y}-\mathbf{y}^{k}\|^2-\|\mathbf{y}^{k-1}-\mathbf{y}^{k}\|^2\right],\ \forall \mathbf{z}\in X^m\times\mathbb{R}^{md}.$$

Multiplying both sides of the above inequality by $\theta_k$, and summing the resulted inequality from $k=1$ to $N$, we obtain

$$\textstyle\sum_{k=1}^N \theta_k Q(\mathbf{x}^k,\mathbf{y}^k;\mathbf{z}) \le \sum_{k=1}^N \theta_k \Delta_k, \tag{6.1}$$

where

$$\begin{aligned}\Delta_k :=\ & \langle \mathbf{L}(\mathbf{x}^k-\tilde{\mathbf{x}}^k),\mathbf{y}-\mathbf{y}^k\rangle+\eta_k\left[\mathbf{V}(\mathbf{x}^{k-1},\mathbf{x})-\mathbf{V}(\mathbf{x}^k,\mathbf{x})-\mathbf{V}(\mathbf{x}^{k-1},\mathbf{x}^k)\right]\\ &+\tfrac{\tau_k}{2}\left[\|\mathbf{y}-\mathbf{y}^{k-1}\|^2-\|\mathbf{y}-\mathbf{y}^{k}\|^2-\|\mathbf{y}^{k-1}-\mathbf{y}^{k}\|^2\right].\end{aligned} \tag{6.2}$$

Observe that from the definition of $\tilde{\mathbf{x}}^k$ in (3.1), (3.9) and (3.11), we have

$$\begin{aligned}\textstyle\sum_{k=1}^N \theta_k\Delta_k =& \textstyle\sum_{k=1}^N\left[\theta_k\langle \mathbf{L}(\mathbf{x}^k-\mathbf{x}^{k-1}),\mathbf{y}-\mathbf{y}^k\rangle-\alpha_k\theta_k\langle \mathbf{L}(\mathbf{x}^{k-1}-\mathbf{x}^{k-2}),\mathbf{y}-\mathbf{y}^{k-1}\rangle\right]\\
&-\textstyle\sum_{k=1}^N\theta_k\left[\alpha_k\langle \mathbf{L}(\mathbf{x}^{k-1}-\mathbf{x}^{k-2}),\mathbf{y}^{k-1}-\mathbf{y}^k\rangle+\eta_k\mathbf{V}(\mathbf{x}^{k-1},\mathbf{x}^k)+\tfrac{\tau_k}{2}\|\mathbf{y}^{k-1}-\mathbf{y}^k\|^2\right]\\
&+\textstyle\sum_{k=2}^N(\theta_k\eta_k-\theta_{k-1}\eta_{k-1})\mathbf{V}(\mathbf{x}^{k-1},\mathbf{x})+\theta_1\eta_1\mathbf{V}(\mathbf{x}^0,\mathbf{x})-\theta_N\eta_N\mathbf{V}(\mathbf{x}^N,\mathbf{x})\\
&+\textstyle\sum_{k=2}^N(\frac{\theta_k\tau_k}{2}-\frac{\theta_{k-1}\tau_{k-1}}{2})\|\mathbf{y}-\mathbf{y}^{k-1}\|^2+\frac{\theta_1\tau_1}{2}\|\mathbf{y}-\mathbf{y}^0\|^2-\frac{\theta_N\tau_N}{2}\|\mathbf{y}-\mathbf{y}^N\|^2\\
\le& \textstyle\sum_{k=1}^N\left[\theta_k\langle \mathbf{L}(\mathbf{x}^k-\mathbf{x}^{k-1}),\mathbf{y}-\mathbf{y}^k\rangle-\alpha_k\theta_k\langle \mathbf{L}(\mathbf{x}^{k-1}-\mathbf{x}^{k-2}),\mathbf{y}-\mathbf{y}^{k-1}\rangle\right]\\
&-\textstyle\sum_{k=1}^N\theta_k\left[\alpha_k\langle \mathbf{L}(\mathbf{x}^{k-1}-\mathbf{x}^{k-2}),\mathbf{y}^{k-1}-\mathbf{y}^k\rangle+\eta_k\mathbf{V}(\mathbf{x}^{k-1},\mathbf{x}^k)+\tfrac{\tau_k}{2}\|\mathbf{y}^{k-1}-\mathbf{y}^k\|^2\right]\\
&+\theta_1\eta_1\mathbf{V}(\mathbf{x}^0,\mathbf{x})-\theta_N\eta_N\mathbf{V}(\mathbf{x}^N,\mathbf{x})+\tfrac{\theta_1\tau_1}{2}\|\mathbf{y}-\mathbf{y}^0\|^2-\tfrac{\theta_N\tau_N}{2}\|\mathbf{y}-\mathbf{y}^N\|^2\\
\overset{(a)}{\le}& \theta_N\langle \mathbf{L}(\mathbf{x}^N-\mathbf{x}^{N-1}),\mathbf{y}-\mathbf{y}^N\rangle-\theta_N\eta_N\mathbf{V}(\mathbf{x}^{N-1},\mathbf{x}^N)\\
&-\textstyle\sum_{k=2}^N\left[\theta_k\alpha_k\langle \mathbf{L}(\mathbf{x}^{k-1}-\mathbf{x}^{k-2}),\mathbf{y}^{k-1}-\mathbf{y}^k\rangle+\theta_{k-1}\eta_{k-1}\mathbf{V}(\mathbf{x}^{k-2},\mathbf{x}^{k-1})+\tfrac{\theta_k\tau_k}{2}\|\mathbf{y}^{k-1}-\mathbf{y}^k\|^2\right]\\
&+\theta_1\eta_1\mathbf{V}(\mathbf{x}^0,\mathbf{x})-\theta_N\eta_N\mathbf{V}(\mathbf{x}^N,\mathbf{x})+\tfrac{\theta_1\tau_1}{2}\|\mathbf{y}-\mathbf{y}^0\|^2-\tfrac{\theta_N\tau_N}{2}\|\mathbf{y}-\mathbf{y}^N\|^2\\
\overset{(b)}{\le}& \theta_N\langle \mathbf{L}(\mathbf{x}^N-\mathbf{x}^{N-1}),\mathbf{y}-\mathbf{y}^N\rangle-\theta_N\eta_N\mathbf{V}(\mathbf{x}^{N-1},\mathbf{x}^N)\\
&+\textstyle\sum_{k=2}^N\left(\frac{\theta_{k-1}\alpha_k\|\mathbf{L}\|^2}{2\tau_k}-\frac{\theta_{k-1}\eta_{k-1}}{2}\right)\|\mathbf{x}^{k-2}-\mathbf{x}^{k-1}\|^2\\
&+\theta_1\eta_1\mathbf{V}(\mathbf{x}^0,\mathbf{x})-\theta_N\eta_N\mathbf{V}(\mathbf{x}^N,\mathbf{x})+\tfrac{\theta_1\tau_1}{2}\|\mathbf{y}-\mathbf{y}^0\|^2-\tfrac{\theta_N\tau_N}{2}\|\mathbf{y}-\mathbf{y}^N\|^2\\
\overset{(c)}{\le}& \theta_N\langle \mathbf{L}(\mathbf{x}^N-\mathbf{x}^{N-1}),\mathbf{y}-\mathbf{y}^N\rangle-\theta_N\eta_N\mathbf{V}(\mathbf{x}^{N-1},\mathbf{x}^N)\\
&+\theta_1\eta_1\mathbf{V}(\mathbf{x}^0,\mathbf{x})-\theta_N\eta_N\mathbf{V}(\mathbf{x}^N,\mathbf{x})+\tfrac{\theta_1\tau_1}{2}\|\mathbf{y}-\mathbf{y}^0\|^2-\tfrac{\theta_N\tau_N}{2}\|\mathbf{y}-\mathbf{y}^N\|^2\\
\overset{(d)}{\le}& \theta_N\langle \mathbf{y}^N,\mathbf{L}(\mathbf{x}^{N-1}-\mathbf{x}^N)\rangle-\theta_N\eta_N\mathbf{V}(\mathbf{x}^{N-1},\mathbf{x}^N)-\tfrac{\theta_1\tau_1}{2}\|\mathbf{y}^N\|^2\\
&+\theta_1\eta_1\mathbf{V}(\mathbf{x}^0,\mathbf{x})+\tfrac{\theta_1\tau_1}{2}\|\mathbf{y}^0\|^2+\langle \mathbf{y},\theta_N\mathbf{L}(\mathbf{x}^N-\mathbf{x}^{N-1})+\theta_1\tau_1(\mathbf{y}^N-\mathbf{y}^0)\rangle,\\
\overset{(e)}{\le}& \left(\tfrac{\theta_N\|\mathbf{L}\|^2}{2\eta_N}-\tfrac{\theta_1\tau_1}{2}\right)\|\mathbf{y}^N\|^2+\theta_1\eta_1\mathbf{V}(\mathbf{x}^0,\mathbf{x})+\tfrac{\theta_1\tau_1}{2}\|\mathbf{y}^0\|^2+\langle \mathbf{y},\theta_N\mathbf{L}(\mathbf{x}^N-\mathbf{x}^{N-1})+\theta_1\tau_1(\mathbf{y}^N-\mathbf{y}^0)\rangle,\end{aligned} \tag{6.3}$$

where (a) follows from (3.10) and the fact that $\mathbf{x}^{-1}=\mathbf{x}^0$, (b) follows from the simple relation that $b\langle u,v\rangle-a\|v\|^2/2\le b^2\|u\|^2/(2a),\forall a>0$, (3.10) and (2.6), (c) follows from (3.12), (d) follows from (3.13), $\|\mathbf{y}-\mathbf{y}^0\|^2-\|\mathbf{y}-\mathbf{y}^N\|^2=\|\mathbf{y}^0\|^2-\|\mathbf{y}^N\|^2-2\langle \mathbf{y},\mathbf{y}^0-\mathbf{y}^N\rangle$ and arranging the terms accordingly, (e) follows from (2.6) and the relation $b\langle u,v\rangle-a\|v\|^2/2\le b^2\|u\|^2/(2a),\forall a>0$. The desired result in (3.15) then follows from this relation, (3.14), (6.1) and the convexity of $Q$.

Furthermore, from (6.3)(c), (2.6) and the fact that $\sum_{k=1}^N\theta_k Q(\mathbf{x}^k,\mathbf{y}^k;\mathbf{z}^*)\ge 0$, if we fix $\mathbf{z}=\mathbf{z}^*=(\mathbf{x}^*,\mathbf{y}^*)$ in the above relation, we have

$$\tfrac{\theta_N\tau_N}{2}\|\mathbf{x}^{N-1}-\mathbf{x}^N\|^2\le\theta_N\langle \mathbf{L}(\mathbf{x}^N-\mathbf{x}^{N-1}),\mathbf{y}^*-\mathbf{y}^N\rangle-\tfrac{\theta_N\tau_N}{2}\|\mathbf{y}^*-\mathbf{y}^N\|^2+\theta_1\eta_1\mathbf{V}(\mathbf{x}^0,\mathbf{x}^*)+\tfrac{\theta_1\tau_1}{2}\|\mathbf{y}^*-\mathbf{y}^0\|^2$$

$$\leq \tfrac{\theta_N\|\mathbf{L}\|^2}{2\tau_N}\|\mathbf{x}^{N-1}-\mathbf{x}^N\|^2+\theta_1\eta_1\mathbf{V}(\mathbf{x}^0,\mathbf{x}^*)+\tfrac{\theta_1\tau_1}{2}\|\mathbf{y}^*-\mathbf{y}^0\|^2,$$

Similarly, we obtain

$$\tfrac{\theta_N\tau_N}{2}\|\mathbf{y}^*-\mathbf{y}^N\|^2\leq\tfrac{\theta_N\|\mathbf{L}\|^2}{2\eta_N}\|\mathbf{y}^*-\mathbf{y}^N\|^2+\theta_1\eta_1\mathbf{V}(\mathbf{x}^0,\mathbf{x}^*)+\tfrac{\theta_1\tau_1}{2}\|\mathbf{y}^*-\mathbf{y}^0\|^2,$$

from which the desired result in (3.17) follows. □

Before we provide proofs for the remaining lemmas, we first need to present a result which summarizes an important convergence property of the CS procedure. It needs to be mentioned that the following proposition states a general result holds for CS procedure performed by individual agent $i\in\mathcal{N}$. For notation convenience, we use the notations defined in CS procedure (cf. Algorithm 3).

**Proposition 2** *If $\{\beta_t\}$ and $\{\lambda_t\}$ in the* CS *procedure satisfy*

$$\lambda_{t+1}(\eta\beta_{t+1}-\mu/\mathcal{C})\leq\lambda_t(1+\beta_t)\eta,\ \forall t\geq 1. \tag{6.4}$$

*then, for $t\geq 1$ and $u\in U$,*

$$\begin{aligned}&(\textstyle\sum_{t=1}^T\lambda_t)^{-1}\left[\eta(1+\beta_T)\lambda_T V(u^T,u)+\sum_{t=1}^T\lambda_t\langle\delta^{t-1},u-u^{t-1}\rangle\right]+\Phi(\hat{u}^T)-\Phi(u)\\&\leq(\textstyle\sum_{t=1}^T\lambda_t)^{-1}\left[(\eta\beta_1-\mu/\mathcal{C})\lambda_1V(u^0,u)+\sum_{t=1}^T\frac{(M+\|\delta^{t-1}\|_*)^2\lambda_t}{2\eta\beta_t}\right],\end{aligned} \tag{6.5}$$

*where $\Phi$ is defined as*

$$\Phi(u):=\langle w,u\rangle+\phi(u)+\eta V(x,u) \tag{6.6}$$

*and $\delta^t:=\phi'(u^t)-h^t$.*

*Proof* Noticing that $\phi:=f_i$ in the CS procedure, we have by (1.2)

$$\begin{aligned}\phi(u^t)&\leq\phi(u^{t-1})+\langle\phi'(u^{t-1}),u^t-u^{t-1}\rangle+M\|u^t-u^{t-1}\|\\&=\phi(u^{t-1})+\langle\phi'(u^{t-1}),u-u^{t-1}\rangle+\langle\phi'(u^{t-1}),u^t-u\rangle+M\|u^t-u^{t-1}\|\\&\leq\phi(u)-\tfrac{\mu}{2}\|u-u^{t-1}\|^2+\langle\phi'(u^{t-1}),u^t-u\rangle+M\|u^t-u^{t-1}\|,\end{aligned}$$

where $\phi'(u^{t-1})\in\partial\phi(u^{t-1})$ and $\partial\phi(u^{t-1})$ denotes the subdifferential of $\phi$ at $u^{t-1}$. By applying Lemma 6 to (4.7), we obtain

$$\langle w+h^{t-1},u^t-u\rangle+\eta V(x,u^t)-\eta V(x,u)\leq\eta\beta_tV(u^{t-1},u)-\eta(1+\beta_t)V(u^t,u)-\eta\beta_tV(u^{t-1},u^t),\ \forall u\in U.$$

Combining the above two relations together with (4.25) [2], we conclude that

$$\begin{aligned}&\langle w,u^t-u\rangle+\phi(u^t)-\phi(u)+\langle\delta^{t-1},u-u^{t-1}\rangle+\eta V(x,u^t)-\eta V(x,u)\\&\leq(\eta\beta_t-\mu/\mathcal{C})V(u^{t-1},u)-\eta(1+\beta_t)V(u^t,u)+\langle\delta^{t-1},u^t-u^{t-1}\rangle+M\|u^t-u^{t-1}\|-\eta\beta_tV(u^{t-1},u^t),\ \forall u\in U.\end{aligned} \tag{6.7}$$

Moreover, by Cauchy-Schwarz inequality, (2.3), and the simple fact that $-at^2/2+bt\leq b^2/(2a)$ for any $a>0$, we have

$$\langle\delta^{t-1},u^t-u^{t-1}\rangle+M\|u^t-u^{t-1}\|-\eta\beta_tV(u^{t-1},u^t)\leq(\|\delta^{t-1}\|_*+M)\|u^t-u^{t-1}\|-\tfrac{\eta\beta_t}{2}\|u^t-t^{t-1}\|^2\leq\tfrac{(M+\|\delta^{t-1}\|_*)^2}{2\eta\beta_t}.$$

From the above relation and the definition of $\Phi(u)$ in (6.6), we can rewrite (6.7) as,

$$\Phi(u^t)-\Phi(u)+\langle\delta^{t-1},u-u^{t-1}\rangle\leq(\eta\beta_t-\mu/\mathcal{C})V(u^{t-1},u)-\eta(1+\beta_t)V(u^t,u)+\tfrac{(M+\|\delta^{t-1}\|_*)^2}{2\eta\beta_t},\ \forall u\in U.$$

Multiplying both sides by $\lambda_t$ and summing up the resulting inequalities from $t=1$ to $T$, we obtain

$$\textstyle\sum_{t=1}^T\lambda_t\left[\Phi(u^t)-\Phi(u)+\langle\delta^{t-1},u-u^{t-1}\rangle\right]\leq\sum_{t=1}^T\left[(\eta\beta_t-\mu/\mathcal{C})\lambda_tV(u^{t-1},u)-\eta(1+\beta_t)\lambda_tV(u^t,u)\right]+\sum_{t=1}^T\frac{(M+\|\delta^{t-1}\|_*)^2\lambda_t}{2\eta\beta_t}.$$

[2] Observed that we only need condition (4.25) when $\mu>0$, in other words, the objective functions $f_i$'s are strongly convex.

Hence, in view of (6.4), the convexity of $\Phi$ and the definition of $\hat{u}^T$ in (4.8), we have

$$\begin{aligned}\Phi(\hat{u}^T) - \Phi(u) + (\textstyle\sum_{t=1}^T \lambda_t)^{-1} \sum_{t=1}^T \lambda_t \langle \delta^{t-1}, u - u^{t-1}\rangle \\ \leq (\textstyle\sum_{t=1}^T \lambda_t)^{-1} \left[(\eta\beta_1 - \mu/\mathcal{C})\lambda_1 V(u^0, u) - \eta(1+\beta_T)\lambda_T V(u^T, u) + \sum_{t=1}^T \frac{(M+\|\delta^{t-1}\|_*)^2 \lambda_t}{2\eta\beta_t}\right],\end{aligned}$$

which implies (6.5) immediately.

As a matter of fact, the SDCS method covers the DCS method as a special case when $\delta^t = 0,\ \forall t \geq 0$. Therefore, we investigate the proofs for Lemma 4 and 5 first and then simplify the proofs for Lemma 2 and 3.

We now provide a proof for Lemma 4, which establishes the convergence property of SDCS method for solving general convex problems.

**Proof of Lemma 4**
When $f_i$, $i = 1, \ldots, m$, are general convex functions, we have $\mu = 0$ and $M > 0$ (cf. (1.2)). Therefore, in view of $\phi := f_i$, and $\lambda_t$ and $\beta_t$ defined in (4.11) satisfying condition (6.4) in the CS procedure, equation (6.5) can be rewritten as the following,[3]

$$\begin{aligned}&(\textstyle\sum_{t=1}^T \lambda_t)^{-1} \left[\eta(1+\beta_T)\lambda_T V_i(u_i^T, u_i) + \sum_{t=1}^T \lambda_t \langle \delta_i^{t-1}, u_i - u_i^{t-1}\rangle\right] + \Phi_i(\hat{u}_i^T) - \Phi_i(u_i) \\ &\leq (\textstyle\sum_{t=1}^T \lambda_t)^{-1} \left[\eta\beta_1\lambda_1 V_i(u_i^0, u_i) + \sum_{t=1}^T \frac{(M+\|\delta_i^{t-1}\|_*)^2\lambda_t}{2\eta\beta_t}\right],\ \forall u_i \in X_i.\end{aligned}$$

In view of the above relation, the definition of $\Phi^k$ in (4.9), and the input and output settings in the CS procedure, it is not difficult to see that, for any $k \geq 1$,[4]

$$\begin{aligned}\Phi^k(\hat{\mathbf{x}}^k) - \Phi^k(\mathbf{x}) \ &+ (\textstyle\sum_{t=1}^{T_k} \lambda_t)^{-1} \left[\eta_k(1+\beta_{T_k})\lambda_{T_k}\mathbf{V}(\mathbf{x}^k, \mathbf{x}) + \sum_{t=1}^{T_k}\sum_{i=1}^m \lambda_t \langle \delta_i^{t-1,k}, x_i - u_i^{t-1}\rangle\right] \\ &\leq (\textstyle\sum_{t=1}^{T_k} \lambda_t)^{-1} \left[\eta_k\beta_1\lambda_1\mathbf{V}(\mathbf{x}^{k-1}, \mathbf{x}) + \sum_{t=1}^{T_k}\sum_{i=1}^m \frac{(M+\|\delta_i^{t-1,k}\|_*)^2\lambda_t}{2\eta_k\beta_t}\right],\ \forall \mathbf{x} \in X^m.\end{aligned}$$

By plugging into the above relation the values of $\lambda_t$ and $\beta_t$ in (4.11), together with the definition of $\Phi^k$ in (4.9) and rearranging the terms, we have,

$$\begin{aligned}\langle \mathbf{L}(\hat{\mathbf{x}}^k - \mathbf{x}), \mathbf{y}^k\rangle + F(\hat{\mathbf{x}}^k) - F(\mathbf{x}) &\leq \tfrac{(T_k+1)(T_k+2)\eta_k}{T_k(T_k+3)} \left[\mathbf{V}(\mathbf{x}^{k-1}, \mathbf{x}) - \mathbf{V}(\mathbf{x}^k, \mathbf{x})\right] - \eta_k \mathbf{V}(\mathbf{x}^{k-1}, \hat{\mathbf{x}}^k) \\ &+ \tfrac{2}{T_k(T_k+3)} \textstyle\sum_{t=1}^{T_k}\sum_{i=1}^m \left[(t+1)\langle \delta_i^{t-1,k}, x_i - u_i^{t-1}\rangle + \frac{2(M+\|\delta_i^{t-1,k}\|_*)^2}{\eta_k}\right],\ \forall \mathbf{x} \in X^m.\end{aligned}$$

Moreover, applying Lemma 6 to (4.3), we have, for $k \geq 1$,

$$\langle v_i^k, y_i - y_i^k\rangle \leq \tfrac{\tau_k}{2}\left[\|y_i - y_i^{k-1}\|^2 - \|y_i - y_i^k\|^2 - \|y_i^{k-1} - y_i^k\|^2\right],\ \forall y_i \in \mathbb{R}^d, \tag{6.8}$$

which in view of the definition of $Q$ in (2.7) and the above two relations, then implies that, for $k \geq 1$, $\mathbf{z} \in X^m \times \mathbb{R}^{md}$,

$$\begin{aligned}Q(\hat{\mathbf{x}}^k, \mathbf{y}^k; \mathbf{z}) &= F(\hat{\mathbf{x}}^k) - F(\mathbf{x}) + \langle \mathbf{L}\hat{\mathbf{x}}^k, \mathbf{y}\rangle - \langle \mathbf{L}\mathbf{x}, \mathbf{y}^k\rangle \\ &\leq \langle \mathbf{L}(\hat{\mathbf{x}}^k - \tilde{\mathbf{x}}^k), \mathbf{y} - \mathbf{y}^k\rangle + \tfrac{(T_k+1)(T_k+2)\eta_k}{T_k(T_k+3)} \left[\mathbf{V}(\mathbf{x}^{k-1}, \mathbf{x}) - \mathbf{V}(\mathbf{x}^k, \mathbf{x})\right] \\ &\quad - \eta_k \mathbf{V}(\mathbf{x}^{k-1}, \hat{\mathbf{x}}^k) + \tfrac{\tau_k}{2}\left[\|\mathbf{y} - \mathbf{y}^{k-1}\|^2 - \|\mathbf{y} - \mathbf{y}^k\|^2 - \|\mathbf{y}^{k-1} - \mathbf{y}^k\|^2\right] \\ &\quad + \tfrac{2}{T_k(T_k+3)} \textstyle\sum_{t=1}^{T_k}\sum_{i=1}^m \left[(t+1)\langle \delta_i^{t-1,k}, x_i - u_i^{t-1}\rangle + \frac{2(M+\|\delta_i^{t-1,k}\|_*)^2}{\eta_k}\right].\end{aligned}$$

[3] We added the subscript $i$ to emphasize that this inequality holds for any agent $i \in \mathcal{N}$ with $\phi = f_i$. More specifically, $\Phi_i(u_i) := \langle w_i, u_i\rangle + f_i(u_i) + \eta V_i(x_i, u_i)$.

[4] We added the superscript $k$ in $\delta_i^{t-1,k}$ to emphasize that this error is generated at the $k$-th outer loop.

Multiplying both sides of the above inequality by $\theta_k$, and summing up the resulting inequalities from $k=1$ to $N$, we obtain, for all $\mathbf{z}\in X^m\times\mathbb{R}^{md}$,

$$\textstyle\sum_{k=1}^N\theta_k Q(\hat{\mathbf{x}}^k,\mathbf{y}^k;\mathbf{z})\le\sum_{k=1}^N\theta_k\tilde{\Delta}_k+\sum_{k=1}^N\sum_{t=1}^{T_k}\sum_{i=1}^m\frac{2\theta_k}{T_k(T_k+3)}\left[(t+1)\langle\delta_i^{t-1,k},x_i-u_i^{t-1}\rangle+\frac{2(M+\|\delta_i^{t-1,k}\|_*)^2}{\eta_k}\right], \tag{6.9}$$

where

$$\begin{aligned}\tilde{\Delta}_k:=&\langle\mathbf{L}(\hat{\mathbf{x}}^k-\tilde{\mathbf{x}}^k),\mathbf{y}-\mathbf{y}^k\rangle+\tfrac{(T_k+1)(T_k+2)\eta_k}{T_k(T_k+3)}\left[\mathbf{V}(\mathbf{x}^{k-1},\mathbf{x})-\mathbf{V}(\mathbf{x}^k,\mathbf{x})\right]\\&-\eta_k\mathbf{V}(\mathbf{x}^{k-1},\hat{\mathbf{x}}^k)+\tfrac{\tau_k}{2}\left[\|\mathbf{y}-\mathbf{y}^{k-1}\|^2-\|\mathbf{y}-\mathbf{y}^k\|^2-\|\mathbf{y}^{k-1}-\mathbf{y}^k\|^2\right].\end{aligned} \tag{6.10}$$

Since $\tilde{\Delta}_k$ in (6.10) shares a similar structure with $\Delta_k$ in (6.2) (with $\mathbf{x}^k$ in the first and the fourth terms being replaced by $\hat{\mathbf{x}}^k$), we can follow the procedure in (6.3) to simplify the RHS of (6.9). The only difference is in the coefficient of the term $[\mathbf{V}(\mathbf{x}^{k-1},\mathbf{x})-\mathbf{V}(\mathbf{x}^k,\mathbf{x})]$. Hence, by using condition (4.10) in place of (3.9), we obtain

$$\begin{aligned}\textstyle\sum_{k=1}^N\theta_k Q(\hat{\mathbf{x}}^k,\mathbf{y}^k;\mathbf{z})&\le\sum_{k=1}^N\theta_k\tilde{\Delta}_k+\sum_{k=1}^N\sum_{t=1}^{T_k}\sum_{i=1}^m\frac{2\theta_k}{T_k(T_k+3)}\left[(t+1)\langle\delta_i^{t-1,k},x_i-u_i^{t-1}\rangle+\frac{2(M+\|\delta_i^{t-1,k}\|_*)^2}{\eta_k}\right]\\&\le\tfrac{(T_1+1)(T_1+2)\theta_1\eta_1}{T_1(T_1+3)}\mathbf{V}(\mathbf{x}^0,\mathbf{x})+\tfrac{\theta_1\tau_1}{2}\|\mathbf{y}^0\|^2+\langle\hat{\mathbf{s}},\mathbf{y}\rangle\\&\quad+\textstyle\sum_{k=1}^N\sum_{t=1}^{T_k}\sum_{i=1}^m\frac{2\theta_k}{T_k(T_k+3)}\left[(t+1)\langle\delta_i^{t-1,k},x_i-u_i^{t-1}\rangle+\frac{4(M^2+\|\delta_i^{t-1,k}\|_*^2)}{\eta_k}\right],\ \forall\mathbf{z}\in X^m\times\mathbb{R}^{md},\end{aligned} \tag{6.11}$$

where

$$\hat{\mathbf{s}}:=\theta_N\mathbf{L}(\hat{\mathbf{x}}^N-\mathbf{x}^{N-1})+\theta_1\tau_1(\mathbf{y}^N-\mathbf{y}^0). \tag{6.12}$$

Our result in (5.5) immediately follows from the convexity of $Q$.

Furthermore, in view of (6.3)(c) and (6.9), we can obtain the following similar result (with $\mathbf{x}^N$ in the first and the second terms of the RHS being replaced with $\hat{\mathbf{x}}^N$),

$$\begin{aligned}\textstyle\sum_{k=1}^N\theta_k Q(\hat{\mathbf{x}}^k,\mathbf{y}^k;\mathbf{z})&\le\theta_N\langle\mathbf{L}(\hat{\mathbf{x}}^N-\mathbf{x}^{N-1}),\mathbf{y}-\mathbf{y}^N\rangle-\theta_N\eta_N\mathbf{V}(\mathbf{x}^{N-1},\hat{\mathbf{x}}^N)\\&\quad+\tfrac{(T_1+1)(T_1+2)\theta_1\eta_1}{T_1(T_1+3)}\mathbf{V}(\mathbf{x}^0,\mathbf{x})-\tfrac{(T_N+1)(T_N+2)\theta_N\eta_N}{T_N(T_N+3)}\mathbf{V}(\mathbf{x}^N,\mathbf{x})+\tfrac{\theta_1\tau_1}{2}\|\mathbf{y}-\mathbf{y}^0\|^2-\tfrac{\theta_N\tau_N}{2}\|\mathbf{y}-\mathbf{y}^N\|^2\\&\quad+\textstyle\sum_{k=1}^N\sum_{t=1}^{T_k}\sum_{i=1}^m\frac{\theta_k}{T_k(T_k+3)}\left[(t+1)\langle\delta_i^{t-1,k},x_i-u_i^{t-1}\rangle+\frac{4(M^2+\|\delta_i^{t-1,k}\|_*^2)}{\eta_k}\right].\end{aligned}$$

Therefore, in view of the fact that $\sum_{k=1}^N\theta_k Q(\hat{\mathbf{x}}^k,\mathbf{y}^k;\mathbf{z}^*)\ge 0$ for any saddle point $\mathbf{z}^*=(\mathbf{x}^*,\mathbf{y}^*)$ of (1.8), and (2.6), by fixing $\mathbf{z}=\mathbf{z}^*$ and rearranging terms, we obtain

$$\begin{aligned}\tfrac{\theta_N\eta_N}{2}\|\hat{\mathbf{x}}^N-\mathbf{x}^{N-1}\|^2&\le\theta_N\langle\mathbf{L}(\hat{\mathbf{x}}^N-\mathbf{x}^{N-1}),\mathbf{y}^*-\mathbf{y}^N\rangle-\tfrac{\theta_N\tau_N}{2}\|\mathbf{y}^*-\mathbf{y}^N\|^2\\&\quad+\tfrac{(T_1+1)(T_1+2)\theta_1\eta_1}{T_1(T_1+3)}\mathbf{V}(\mathbf{x}^0,\mathbf{x}^*)+\tfrac{\theta_1\tau_1}{2}\|\mathbf{y}^*-\mathbf{y}^0\|^2\\&\quad+\textstyle\sum_{k=1}^N\sum_{t=1}^{T_k}\sum_{i=1}^m\frac{2\theta_k}{T_k(T_k+3)}\left[(t+1)\langle\delta_i^{t-1,k},x_i^*-u_i^{t-1}\rangle+\frac{4(M^2+\|\delta_i^{t-1,k}\|_*^2)}{\eta_k}\right]\\&\le\tfrac{\theta_N\|\mathbf{L}\|^2}{2\tau_N}\|\hat{\mathbf{x}}^N-\mathbf{x}^{N-1}\|^2+\tfrac{(T_1+1)(T_1+2)\theta_1\eta_1}{T_1(T_1+3)}\mathbf{V}(\mathbf{x}^0,\mathbf{x}^*)+\tfrac{\theta_1\tau_1}{2}\|\mathbf{y}^*-\mathbf{y}^0\|^2\\&\quad+\textstyle\sum_{k=1}^N\sum_{t=1}^{T_k}\sum_{i=1}^m\frac{2\theta_k}{T_k(T_k+3)}\left[(t+1)\langle\delta_i^{t-1,k},x_i^*-u_i^{t-1}\rangle+\frac{4(M^2+\|\delta_i^{t-1,k}\|_*^2)}{\eta_k}\right],\end{aligned} \tag{6.13}$$

where the second inequality follows from the relation $b\langle u,v\rangle-a\|v\|^2/2\le b^2\|u\|^2/(2a),\forall a>0$.

Similarly, we obtain

$$\tfrac{\theta_N\tau_N}{2}\|\mathbf{y}^*-\mathbf{y}^N\|^2\le\tfrac{\theta_N\|\mathbf{L}\|^2}{2\eta_N}\|\mathbf{y}^*-\mathbf{y}^N\|^2+\tfrac{(T_1+1)(T_1+2)\theta_1\eta_1}{T_1(T_1+3)}\mathbf{V}(\mathbf{x}^0,\mathbf{x}^*)+\tfrac{\theta_1\tau_1}{2}\|\mathbf{y}^*-\mathbf{y}^0\|^2 \tag{6.14}$$

$$+\sum_{k=1}^{N}\sum_{t=1}^{T_k}\sum_{i=1}^{m}\frac{2\theta_k}{T_k(T_k+3)}\left[(t+1)\langle\delta_i^{t-1,k},x_i^*-u_i^{t-1}\rangle+\frac{4(M^2+\|\delta_i^{t-1,k}\|_*^2)}{\eta_k}\right],$$

from which the desired result in (5.6) follows.

The following proof of Lemma 5 establishes the convergence of SDCS method for solving strongly convex problems.

**Proof of Lemma 5**
When $f_i$, $i=1,\ldots,m$, are strongly convex functions, we have $\mu,\ M>0$ (cf. (1.2)). Therefore, in view of Proposition 2 with $\lambda_t$ and $\beta_t$ defined in (4.27) satisfying condition (6.4), the definition of $\Phi^k$ in (4.9), and the input and output settings in the CS procedure, we have for all $k\geq 1$

$$\begin{aligned}\Phi^k(\hat{\mathbf{x}}^k)-\Phi^k(\mathbf{x})\ &+(\textstyle\sum_{t=1}^{T_k}\lambda_t)^{-1}\left[\eta_k(1+\beta_{T_k}^{(k)})\lambda_{T_k}\mathbf{V}(\mathbf{x}^k,\mathbf{x})+\sum_{t=1}^{T_k}\sum_{i=1}^{m}\lambda_t\langle\delta_i^{t-1,k},x_i-u_i^{t-1}\rangle\right]\\ &\leq(\textstyle\sum_{t=1}^{T_k}\lambda_t)^{-1}\left[(\eta_k\beta_1^{(k)}-\mu/\mathcal{C})\lambda_1\mathbf{V}(\mathbf{x}^{k-1},\mathbf{x})+\sum_{t=1}^{T_k}\sum_{i=1}^{m}\frac{(M+\|\delta_i^{t-1,k}\|_*)^2\lambda_t}{2\eta_k\beta_t}\right],\ \forall\mathbf{x}\in X^m.\end{aligned}\tag{6.15}$$

By plugging into the above relation the values of $\lambda_t$ and $\beta_t^{(k)}$ in (4.27), together with the definition of $\Phi^k$ in (4.9) and rearranging the terms, we have

$$\begin{aligned}\langle\mathbf{L}(\hat{\mathbf{x}}^k-\mathbf{x}),\mathbf{y}^k\rangle+F(\hat{\mathbf{x}}^k)-F(\mathbf{x})\ &\leq\eta_k\mathbf{V}(\mathbf{x}^{k-1},\mathbf{x})-(\mu/\mathcal{C}+\eta_k)\mathbf{V}(\mathbf{x}^k,\mathbf{x})-\eta_k\mathbf{V}(\mathbf{x}^{k-1},\hat{\mathbf{x}}^k)\\ &+\tfrac{2}{T_k(T_k+1)}\textstyle\sum_{t=1}^{T_k}\sum_{i=1}^{m}\left[t\langle\delta_i^{t-1,k},x_i-u_i^{t-1}\rangle+\frac{(M+\|\delta_i^{t-1,k}\|_*)^2t}{(t+1)\mu/\mathcal{C}+(t-1)\eta_k}\right],\ \forall\mathbf{x}\in X^m,\ k\geq 1.\end{aligned}$$

In view of (6.8), the above relation and the definition of Q in (2.7), and following the same trick that we used to obtain (6.9), we have, for all $\mathbf{z}\in X^m\times\mathbb{R}^{md}$,

$$\textstyle\sum_{k=1}^{N}\theta_kQ(\hat{\mathbf{x}}^k,\mathbf{y}^k;\mathbf{z})\leq\sum_{k=1}^{N}\theta_k\bar{\Delta}_k+\sum_{k=1}^{N}\sum_{t=1}^{T_k}\sum_{i=1}^{m}\frac{2\theta_k}{T_k(T_k+1)}\left[t\langle\delta_i^{t-1,k},x_i-u_i^{t-1}\rangle+\frac{(M+\|\delta_i^{t-1,k}\|_*)^2t}{(t+1)\mu/\mathcal{C}+(t-1)\eta_k}\right],\tag{6.16}$$

where

$$\begin{aligned}\bar{\Delta}_k:=&\langle\mathbf{L}(\hat{\mathbf{x}}^k-\tilde{\mathbf{x}}^k),\mathbf{y}-\mathbf{y}^k\rangle+\eta_k\mathbf{V}(\mathbf{x}^{k-1},\mathbf{x})-(\mu/\mathcal{C}+\eta_k)\mathbf{V}(\mathbf{x}^k,\mathbf{x})-\eta_k\mathbf{V}(\mathbf{x}^{k-1},\hat{\mathbf{x}}^k)\\ &+\tfrac{\tau_k}{2}\left[\|\mathbf{y}-\mathbf{y}^{k-1}\|^2-\|\mathbf{y}-\mathbf{y}^k\|^2-\|\mathbf{y}^{k-1}-\mathbf{y}^k\|^2\right].\end{aligned}\tag{6.17}$$

Since $\bar{\Delta}_k$ in (6.17) shares a similar structure with $\tilde{\Delta}_k$ in (6.10) (also $\Delta_k$ in (6.2)), we can follow similar procedure as in (6.3) to simplify the RHS of (6.16). Note that the only difference of (6.17) and (6.10) (also (6.2)) is in the coefficient of the terms $\mathbf{V}(\mathbf{x}^{k-1},\mathbf{x})$, and $\mathbf{V}(\mathbf{x}^k,\mathbf{x})$. Hence, by using condition (4.26) in place of (4.10) (also (3.9)) , we obtain $\forall\mathbf{z}\in X^m\times\mathbb{R}^{md}$

$$\begin{aligned}\textstyle\sum_{k=1}^{N}\theta_kQ(\hat{\mathbf{x}}^k,\mathbf{y}^k;\mathbf{z})\leq\ &\theta_1\eta_1\mathbf{V}(\mathbf{x}^0,\mathbf{x})+\tfrac{\theta_1\tau_1}{2}\|\mathbf{y}^0\|^2+\langle\hat{\mathbf{s}},\mathbf{y}\rangle\\ &+\textstyle\sum_{k=1}^{N}\sum_{t=1}^{T_k}\sum_{i=1}^{m}\frac{2\theta_k}{T_k(T_k+1)}\left[t\langle\delta_i^{t-1,k},x_i-u_i^{t-1}\rangle+\frac{2t(M^2+\|\delta_i^{t-1,k}\|_*^2)}{(t+1)\mu/\mathcal{C}+(t-1)\eta_k}\right],\end{aligned}\tag{6.18}$$

where $\hat{\mathbf{s}}$ is defined in (6.12). Our result in (5.14) immediately follows from the convexity of $Q$.

Following the same procedure as we obtain (6.13), for any saddle point $\mathbf{z}^*=(\mathbf{x}^*,\mathbf{y}^*)$ of (1.8), we have

$$\begin{aligned}\tfrac{\theta_N\eta_N}{2}\|\hat{\mathbf{x}}^N-\mathbf{x}^{N-1}\|^2\leq\ &\tfrac{\theta_N\|\mathbf{L}\|^2}{2\tau_N}\|\mathbf{x}^N-\mathbf{x}^{N-1}\|^2+\theta_1\eta_1\mathbf{V}(\mathbf{x}^0,\mathbf{x}^*)+\tfrac{\theta_1\tau_1}{2}\|\mathbf{y}^*-\mathbf{y}^0\|^2\\ &+\textstyle\sum_{k=1}^{N}\sum_{t=1}^{T_k}\sum_{i=1}^{m}\frac{2\theta_k}{T_k(T_k+1)}\left[t\langle\delta_i^{t-1,k},x_i^*-u_i^{t-1}\rangle+\frac{2t(M^2+\|\delta_i^{t-1,k}\|_*^2)}{(t+1)\mu/\mathcal{C}+(t-1)\eta_k}\right],\\ \tfrac{\theta_N\tau_N}{2}\|\mathbf{y}^*-\mathbf{y}^N\|^2\leq\ &\tfrac{\theta_N\|\mathbf{L}\|^2}{2\eta_N}\|\mathbf{y}^*-\mathbf{y}^N\|^2+\theta_1\eta_1\mathbf{V}(\mathbf{x}^0,\mathbf{x}^*)+\tfrac{\theta_1\tau_1}{2}\|\mathbf{y}^*-\mathbf{y}^0\|^2\\ &+\textstyle\sum_{k=1}^{N}\sum_{t=1}^{T_k}\sum_{i=1}^{m}\frac{2\theta_k}{T_k(T_k+1)}\left[t\langle\delta_i^{t-1,k},x_i^*-u_i^{t-1}\rangle+\frac{2t(M^2+\|\delta_i^{t-1,k}\|_*^2)}{(t+1)\mu/\mathcal{C}+(t-1)\eta_k}\right],\end{aligned}\tag{6.19}$$

from which the desired result in (5.15) follows.

We are ready to provide proofs for Lemma 2 and 3, which demonstrates the convergence properties of the deterministic communication sliding method.

**Proof of Lemma 2**
When $f_i,\ i=1,\ldots,m$ are general nonsmooth convex functions, we have $\delta_i^t=0$, $\mu=0$ and $M>0$. Therefore, in view of (6.11), we have, $\forall \mathbf{z}\in X^m\times\mathbb{R}^{md}$,

$$\textstyle\sum_{k=1}^N\theta_k Q(\hat{\mathbf{x}}^k,\mathbf{y}^k;\mathbf{z})\le \frac{(T_1+1)(T_1+2)\theta_1\eta_1}{T_1(T_1+3)}\mathbf{V}(\mathbf{x}^0,\mathbf{x})+\frac{\theta_1\tau_1}{2}\|\mathbf{y}^0\|^2+\langle\hat{\mathbf{s}},\mathbf{y}\rangle+\sum_{k=1}^N\frac{4mM^2\theta_k}{(T_k+3)\eta_k},$$

where $\hat{\mathbf{s}}$ is defined in (6.12). Our result in (4.12) immediately follows from the convexity of $Q$. Moreover, our result in (4.13) follows from setting $\delta_i^{t-1,k}=0$ in (6.13) and (6.14).

**Proof of Lemma 3**
When $f_i,\ i=1,\ldots,m$ are strongly convex functions, we have $\delta_i^t=0$ and $\mu,\ M>0$. Therefore, in view of (6.18), we obtain, $\forall \mathbf{z}\in X^m\times\mathbb{R}^{md}$,

$$\textstyle\sum_{k=1}^N\theta_k Q(\hat{\mathbf{x}}^k,\mathbf{y}^k;\mathbf{z})\le \theta_1\eta_1\mathbf{V}(\mathbf{x}^0,\mathbf{x})+\frac{\theta_1\tau_1}{2}\|\mathbf{y}^0\|^2+\langle\hat{\mathbf{s}},\mathbf{y}\rangle+\sum_{k=1}^N\sum_{t=1}^{T_k}\frac{2mM^2\theta_k}{T_k(T_k+1)}\frac{t}{(t+1)\mu/\mathcal{C}+(t-1)\eta_k},$$

where $\hat{\mathbf{s}}$ is defined in (6.12). Our result in (4.28) immediately follows from the convexity of $Q$. Also, the result in (4.29) follows by setting $\delta_i^{t-1,k}=0$ in (6.19).

**Proof of Theorem 7**
Observe that by Assumption (5.1), (5.2) and (5.23) on the SO and the definition of $u_i^{t,k}$, the sequence $\{\langle\delta_i^{t-1,k},x_i^*-u_i^{t-1,k}\rangle\}_{1\le i\le m,1\le t\le T_k,k\ge1}$ is a martingale-difference sequence. Denoting

$$\gamma_{k,t}:=\frac{\theta_k\lambda_t}{\sum_{t=1}^{T_k}\lambda_t},$$

and using the large-deviation theorem for martingale-difference sequence (e.g. Lemma 2 of [25]) and the fact that

$$\begin{aligned}&\mathbb{E}[\exp\{\gamma_{k,t}^2\langle\delta_i^{t-1,k},x_i^*-u_i^{t-1,k}\rangle^2/(2\gamma_{k,t}^2\bar{V}_i(x_i^*)\sigma^2)\}]\\&\quad\le\mathbb{E}[\exp\{\|\delta_i^{t-1,k}\|_*^2,\|x_i^*-u_i^{t-1,k}\|^2/(2\bar{V}_i(x_i^*)\sigma^2)\}]\\&\quad\le\mathbb{E}[\exp\{\|\delta_i^{t-1,k}\|_*^2/\sigma^2\}]\le\exp\{1\},\end{aligned}$$

we conclude that, $\forall\zeta>0$,

$$\text{Prob}\left\{\textstyle\sum_{k=1}^N\sum_{t=1}^{T_k}\sum_{i=1}^m\gamma_{k,t}\langle\delta_i^{t-1,k},u_i^{t-1,k}-x_i^*\rangle>\zeta\sigma\sqrt{2\bar{\mathbf{V}}(\mathbf{x}^*)\sum_{k=1}^N\sum_{t=1}^{T_k}\gamma_{k,t}^2}\right\}\le\exp\{-\zeta^2/3\}.\tag{6.20}$$

Now let

$$S_{k,t}:=\frac{\theta_k\lambda_t}{\left(\sum_{t=1}^{T_k}\lambda_t\right)\eta_k\beta_t},$$

and $S:=\sum_{k=1}^N\sum_{t=1}^{T_k}\sum_{i=1}^m S_{k,t}$. By the convexity of exponential function, we have

$$\mathbb{E}[\exp\{\tfrac{1}{S}\textstyle\sum_{k=1}^N\sum_{t=1}^{T_k}\sum_{i=1}^m S_{k,t}\|\delta_i^{t-1,k}\|_*^2/\sigma^2\}]\le\mathbb{E}[\tfrac{1}{S}\sum_{k=1}^N\sum_{t=1}^{T_k}\sum_{i=1}^m S_{k,t}\exp\{\|\delta_i^{t-1,k}\|_*^2/\sigma^2\}]\le\exp\{1\},$$

where the last inequality follows from Assumption (5.23). Therefore, by Markov's inequality, for all $\zeta>0$,

$$\begin{aligned}&\text{Prob}\left\{\textstyle\sum_{k=1}^N\sum_{t=1}^{T_k}\sum_{i=1}^m S_{k,t}\|\delta_i^{t-1,k}\|_*^2>(1+\zeta)\sigma^2\sum_{k=1}^N\sum_{t=1}^{T_k}\sum_{i=1}^m S_{k,t}\right\}\\&\quad=\text{Prob}\left\{\exp\left\{\tfrac{1}{S}\textstyle\sum_{k=1}^N\sum_{t=1}^{T_k}\sum_{i=1}^m S_{k,t}\|\delta_i^{t-1,k}\|_*^2/\sigma^2\right\}\ge\exp\{1+\zeta\}\right\}\le\exp\{-\zeta\}.\end{aligned}\tag{6.21}$$

Combing (6.20), (6.21), (5.5) and (2.9), our result in (5.25) immediately follows.

## 7 Concluding Remarks

In this paper, we present a new class of decentralized primal-dual methods which can significantly reduce the number of inter-node communications required to solve the distributed optimization problem in (1.1). More specifically, we show that by using these algorithms, the total number of communication rounds can be significantly reduced to $\mathcal{O}(1/\epsilon)$ when the objective functions $f_i$'s are convex and not necessarily smooth. By properly designing the communication sliding algorithms, we demonstrate that the $\mathcal{O}(1/\epsilon)$ number of communications can still be maintained for general convex objective functions (and it can be further reduced to $\mathcal{O}(1/\sqrt{\epsilon})$ for strongly convex objective functions) even if the local subproblems are solved inexactly through iterative procedure (cf. CS procedure) by the network agents. In this case, the number of intra-node subgradient computations that we need will be bounded by $\mathcal{O}(1/\epsilon^2)$ (resp., $\mathcal{O}(1/\epsilon)$) when the objective functions $f_i$'s are convex (resp., strongly convex), which is comparable to that required in centralized nonsmooth optimization and not improvable in general. We also establish similar complexity bounds for solving stochastic decentralized optimization counterpart by developing the stochastic communication sliding methods, which can provide communication-efficient ways to deal with streaming data and decentralized statistical inference. All these decentralized communication sliding algorithms have the potential to significantly increase the performance of multiagent systems, where the bottleneck exists in the communication.

## References


1. K. Arrow, L. Hurwicz, and H. Uzawa. *Studies in Linear and Non-linear Programming*. Stanford Mathematical Studies in the Social Sciences. Stanford University Press, 1958.
2. D. P. Bertsekas. Incremental proximal methods for large scale convex optimization. *Mathematical Programming*, 129:163–195, 2011.
3. D. P. Bertsekas. Incremental aggregated proximal and augmented lagrangian algorithms. Technical Report LIDS-P-3176, Laboratory for Information and Decision Systems, 2015.
4. S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, January 2011.
5. L.M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7(3):200 – 217, 1967.
6. A. Chambolle and T. Pock. On the ergodic convergence rates of a first-order primal-dual algorithm. Oct. 30, 2014.
7. Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vis.*, 40(1):120–145, May 2011.
8. T. Chang and M. Hong. Stochastic proximal gradient consensus over random networks. http://arxiv.org/abs/1511.08905, 2015.
9. T. Chang, M. Hong, and X. Wang. Multi-agent distributed optimization via inexact consensus admm. http://arxiv.org/abs/1402.6065, 2014.
10. T.-H. Chang, A. Nedić, and A. Scaglione. Distributed constrained optimization by consensus-based primal-dual perturbation method. *Automatic Control, IEEE Transactions on*, 59(6):1524–1538, June 2014.
11. A. Chen and A. Ozdaglar. A fast distributed proximal gradient method. In *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*, pages 601–608, Oct 2012.
12. Y. Chen, G. Lan, and Y. Ouyang. Optimal primal-dual methods for a class of saddle point problems. 24(4):1779–1814, 2014.
13. C. Dang and G. Lan. Randomized first-order methods for saddle point optimization. Technical Report 32611, Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL, 2015.
14. J. Duchi, A. Agarwal, and M. Wainwright. Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Trans. Automat. Contr.*, 57(3):592–606, 2012.
15. J. W. Durham, A. Franchi, and F. Bullo. Distributed pursuit-evasion without mapping or global localization via local frontiers. *Autonomous Robots*, 32(1):81–95, 2012.
16. S. Ghadimi and G. Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization I: A generic algorithmic framework. *SIAM Journal on Optimization*, 22(4):1469–1492, 2012.
17. S. Ghadimi and G. Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization, ii: shrinking procedures and optimal algorithms. *SIAM Journal on Optimization*, 23(4):2061–2089, 2013.
18. M. Gurbuzbalaban, A. Ozdaglar, and P. Parrilo. On the convergence rate of incremental aggregated gradient algorithms. http://arxiv.org/abs/1506.02081, 2015.
19. B. He and X. Yuan. On the o(1/n) convergence rate of the douglas-rachford alternating direction method. *SIAM Journal on Numerical Analysis*, 50(2):700–709, 2012.
20. N. He, A. Juditsky, and A. Nemirovski. Mirror prox algorithm for multi-term composite minimization and semi-separable problems. *Journal of Computational Optimization and Applications*, 103:127–152, 2015.
21. A. Jadbabaie, Jie Lin, and A.S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988 – 1001, June 2003.
22. D. Jakovetic, J. Xavier, and J. Moura. Fast distributed gradient methods. *Automatic Control, IEEE Transactions on*, 59(5):1131–1145, May 2014.

23. G. Lan. An optimal method for stochastic composite optimization. *Mathematical Programming*, 133(1):365–397, 2012.
24. G. Lan. Gradient sliding for composite optimization. *Mathematical Programming*, 159(1):201–235, 2016.
25. G. Lan, A. Nemirovski, and A. Shapiro. Validation analysis of mirror descent stochastic approximation method. *Math. Program.*, 134(2):425–458, 2012.
26. G. Lan and Y. Zhou. An optimal randomized incremental gradient method. http://arxiv.org/abs/1507.02000, 2015.
27. I. Lobel and A. Ozdaglar. Distributed subgradient methods for convex optimization over random networks. *IEEE Transactions on Automatic Control*, 56(6):1291 –1306, June 2011.
28. A. Makhdoumi and A. Ozdaglar. Convergence rate of distributed admm over networks. http://arxiv.org/abs/1601.00194, 2016.
29. A. Mokhtari, W. Shi, Q. Ling, and A. Ribeiro. Dqm: Decentralized quadratically approximated alternating direction method of multipliers. http://arxiv.org/abs/1508.02073, 2015.
30. A. Mokhtari, W. Shi, Q. Ling, and A. Ribeiro. A decentralized second-order method with exact linear convergence rate for consensus optimization. http://arxiv.org/abs/1602.00596, 2016.
31. R. D. C. Monteiro and B. F. Svaiter. On the complexity of the hybrid proximal extragradient method for the iterates and the ergodic mean. *SIAM Journal on Optimization*, 20(6):2755–2787, 2010.
32. R. D. C. Monteiro and B. F. Svaiter. Complexity of variants of tseng's modified f-b splitting and korpelevich's methods for hemivariational inequalities with applications to saddle-point and convex optimization problems. *SIAM Journal on Optimization*, 21(4):1688–1720, 2011.
33. R. D. C. Monteiro and B. F. Svaiter. Iteration-complexity of block-decomposition algorithms and the alternating direction method of multipliers. *SIAM Journal on Optimization*, 23(1):475–507, 2013.
34. R.D.C. Monteiro and B.F. Svaiter. On the complexity of the hybrid proximal projection method for the iterates and the ergodic mean. 20:2755–2787, 2010.
35. A. Nedić. Asynchronous broadcast-based convex optimization over a network. *IEEE Trans. Automat. Contr.*, 56(6):1337–1351, 2011.
36. A. Nedić, D. P. Bertsekas, and V. S. Borkar. Distributed asynchronous incremental subgradient methods. *Inherently Parallel Algorithms in Feasibility and Optimization and Their Applications*, pages 311–407, 2001.
37. A. Nedić and A. Olshevsky. Distributed optimization over time-varying directed graphs. *IEEE Transactions on Automatic Control*, 60(3):601–615, March 2015.
38. A. Nedić, A. Olshevsky, and W. Shi. Achieving geometric convergence for distributed optimization over time-varying graphs. http://arxiv.org/abs/1607.03218, 2016.
39. A. Nedić and A. Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.
40. A. S. Nemirovski. Prox-method with rate of convergence $o(1/t)$ for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. 15:229–251, 2005.
41. A. S. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. 19:1574–1609, 2009.
42. A. S. Nemirovski and D. Yudin. *Problem complexity and method efficiency in optimization*. Wiley-Interscience Series in Discrete Mathematics. John Wiley, XV, 1983.
43. Y. E. Nesterov. Smooth minimization of nonsmooth functions. *Mathematical Programming*, 61(2):275–319, 2015.
44. Y. Ouyang, Y. Chen, G. Lan, and E. Pasiliao Jr. An accelerated linearized alternating direction method of multipliers. *SIAM Journal on Imaging Sciences*, 8(1):644–681, 2015.
45. G. Qu and N. Li. Harnessing smoothness to accelerate distributed optimization. http://arxiv.org/abs/1605.07112, 2016.
46. M. Rabbat. Multi-agent mirror descent for decentralized stochastic optimization. In *2015 IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 517–520, Dec 2015.
47. M. Rabbat and R. D. Nowak. Distributed optimization in sensor networks. In *IPSN*, pages 20–27, 2004.
48. S. S. Ram, A. Nedić, and V. V. Veeravalli. Incremental stochastic subgradient algorithms for convex optimization. *SIAM J. on Optimization*, 20(2):691–717, June 2009.
49. S. S. Ram, A. Nedić, and V. V. Veeravalli. Distributed Stochastic Subgradient Projection Algorithms for Convex Optimization. *Journal of Optimization Theory and Applications*, 147:516–545, 2010.
50. S. S. Ram, V. V. Veeravalli, and A. Nedić. Distributed non-autonomous power control through distributed convex optimization. In *IEEE INFOCOM*, pages 3001–3005, 2009.
51. W. Shi, Q. Ling, G. Wu, and W. Yin. On the linear convergence of the admm in decentralized consensus optimization. *IEEE Transactions on Signal Processing*, 62(7):1750–1761, 2014.
52. W. Shi, Q. Ling, G. Wu, and W. Yin. Extra: An exact first-order algorithm for decentralized consensus optimization. *SIAM Journal on Optimization*, 25(2):944–966, 2015.
53. W. Shi, Q. Ling, G. Wu, and W. Yin. A proximal gradient algorithm for decentralized composite optimization. *IEEE Transactions on Signal Processing*, 63(22):6013–6023, November 2015.
54. A. Simonetto, L. Kester, and G. Leus. Distributed time-varying stochastic optimization and utility-based communication. http://arxiv.org/abs/1408.5294, 2014.
55. H. Terelius, U. Topcu, and R. Murray. Decentralized multi-agent optimization via dual decomposition. *IFAC Proceedings Volumes*, 44(1):11245–11251, 2011.
56. K. Tsianos, S. Lawlor, and M. Rabbat. Consensus-based distributed optimization: Practical issues and applications in large-scale machine learning. In *Proceedings of the 50th Allerton Conference on Communication, Control, and Computing*, 2012.
57. K. Tsianos, S. Lawlor, and M. Rabbat. Push-sum distributed dual-averaging for convex optimization. In *Proceedings of the 51st IEEE Conference on Decision and Control*, pages 5453–5458, Maui, Hawaii, December 2012.
58. K. Tsianos and M. Rabbat. Consensus-based distributed online prediction and optimization. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 807–810, Dec 2013.

59. J. Tsitsiklis, D. Bertsekas, and M. Athans. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Transactions on Automatic Control*, 31(9):803 – 812, Sep. 1986.
60. J. N. Tsitsiklis. *Problems in Decentralized Decision Making and Computation.* PhD thesis, Massachusetts Inst. Technol., Cambridge, MA, 1984.
61. M. Wang and D. P. Bertsekas. Incremental constraint projection-proximal methods for nonsmooth convex optimization. Technical Report LIDS-P-2907, Laboratory for Information and Decision Systems, 2013.
62. E. Wei and A. Ozdaglar. On the $O(1/k)$ convergence of asynchronous distributed alternating direction method of multipliers. http://arxiv.org/pdf/1307.8254, 2013.
63. C. Xi, Q. Wu, and U. A. Khan. Distributed mirror descent over directed graphs. http://arxiv.org/abs/1412.5526, 2014.
64. Y. Zhang and L. Xiao. Stochastic primal-dual coordinate method for regularized empirical risk minimization. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 353–361, 2015.
65. M. Zhu and S. Martinez. On distributed convex optimization under inequality and equality constraints. *Automatic Control, IEEE Transactions on*, 57(1):151–164, Jan 2012.