J Syst Sci Complex (2023) 36: 577–590

Primal-Dual ε-Subgradient Method for Distributed Optimization*

ZHU Kui · TANG Yutao

DOI: 10.1007/s11424-023-1321-y
Received: 26 August 2021 / Revised: 15 December 2021

Abstract This paper studies the distributed optimization problem when the objective functions might be nondifferentiable and subject to heterogeneous set constraints. Unlike existing subgradient methods, the authors focus on the case when the exact subgradients of the local objective functions can not be accessed by the agents. To solve this problem, the authors propose a projected primal-dual dynamics using only the objective function's approximate subgradients. The authors first prove that the formulated optimization problem can generally be solved with an error depending upon the accuracy of the available subgradients. Then, the authors show the exact solvability of this distributed optimization problem when the accumulated approximation error of inexact subgradients is not too large. After that, the authors also give a novel componentwise normalized variant to improve the transient behavior of the convergent sequence. The effectiveness of the proposed algorithms is verified by a numerical example.

Keywords Constrained optimization, distributed optimization, ε-subgradient, primal-dual dynamics

1 Introduction

The last decade has witnessed considerable interests in distributed optimization problems due to the numerous applications in signal processing, control, and machine learning. To solve this problem, subgradient information of the objective functions has been widely used due to the cheap iteration cost and well-established convergence properties[1, 2].

Note that most of these subgradient-based results assume the availability of local cost function's exact subgradients. In many circumstances, the function subgradient is computed by solving another auxiliary optimization problem as shown in [3–6]. In practice, we are often only able to solve these subproblems approximately. Hence, in that context, numerical methods solving the original optimization problem are provided with only inexact subgradient information.

ZHU Kui · TANG Yutao (Corresponding author)
School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing 100876, *China.*
Email: yttang@bupt.edu.cn.
*This research was supported by the National Natural Science Foundation of China under Grant No. 61973043.
◇ *This paper was recommended for publication by Editor YOU Keyou.*

This leads us to investigate the solvability of distributed optimization problem using inexact subgradient information.

A close topic is the inexact augmented Lagrangian method. As surveyed in [7], this method has been extensively extended to distributed settings in various ways assuming the primal variables are only obtained in an approximate sense. Nevertheless, most of these results still require the exact gradient or subgradient information of the local objective functions at each given estimate. It is interesting to ask whether the primal-dual method is still effective when only inexact gradient/subgradient information is available.

In this paper, we focus on a typical distributed consensus optimization problem for a sum of convex objective functions subject to heterogeneous set constraints. Although this problem has been partially studied by gradient/subgradient methods in [8–14], its solvability using only inexact subgradient information has not yet been addressed and is still unclear at present. To solve this problem, we convert it into a saddle point seeking problem and present a projected primal-dual ε-subgradient dynamics to deal with the distributedness requirement and set constraints.

The contribution of the paper is summarized as follows:

- We develop a novel primal-dual ε-subgradient method for the distributed consensus optimization problem. When the objective functions are smooth with exact gradients, the proposed algorithms reduce to the primal-dual dynamics considering in [9, 11]. By contrast, the presented results do not require the exact knowledge of the subgradient information of the objective functions. To our knowledge, this might be the first attempt to solve the formulated distributed optimization problem using only inexact subgradients of the local objective functions.

- We investigate the convergence property of the presented method and its suboptimality to solve the formulated problem depending on the accuracy of available subgradients. In particular, we show that if the accumulated error resulting from the subgradient inexactness is not too large, the proposed algorithm under certain diminishing step size will drive all the estimates of agents to reach a consensus about an optimal solution to the global optimization problem. These results extend the conventional ε-subgradient method discussed in [3, 6, 15, 16] to distributed scenarios.

- We also propose a novel componentwise normalized step size as that in [17] to improve the transient performance of our preceding designs. As a byproduct, this normalized step size removes the widely used subgradient boundedness assumption in the literature [8, 9].

The rest of this paper is organized as follows. We first give some preliminaries in Section 2 and then introduce the formulation of our problem in Section 3. Main results are presented in Section 4. After that, we give a numerical example in Section 5 to show the effectiveness of our design. Finally, some concluding remarks are given in Section 6.

2 Preliminary

In this section, we first give some preliminaries about graph theory and convex analysis.

2.1 Graph Theory

Let $\mathbb{R}^n$ be the n-dimensional Euclidean space and $\mathbb{R}^{n\times m}$ be the set of all $n \times m$ matrices. $\mathbf{1}_n$ (or $\mathbf{0}_n$) denotes an n-dimensional all-one (or all-zero) column vector and $\mathbf{1}_{n\times m}$ (or $\mathbf{0}_{n\times m}$) all-one (or all-zero) matrix. $\mathrm{col}(a_1, \cdots, a_n) = [a_1^{\mathrm{T}}, \cdots, a_n^{\mathrm{T}}]^{\mathrm{T}}$ for column vectors $a_1, \cdots, a_n$. For a vector x (or matrix A), $\|x\|$ ($\|A\|$) denotes its Euclidean (or spectral) norm.

A weighted (undirected) graph $\mathcal{G} = (\mathcal{N}, \mathcal{E}, \mathcal{A})$ is defined as follows, where $\mathcal{N} = \{1, \cdots, n\}$ is the set of nodes, $\mathcal{E} \subset \mathcal{N} \times \mathcal{N}$ is the set of edges, and $\mathcal{A} \in \mathbb{R}^{n\times n}$ is a weighted adjacency matrix. $(i, j) \in \mathcal{E}$ denotes an edge leaving from node i and entering node j. The weighted adjacency matrix of this graph $\mathcal{G}$ is described by $\mathcal{A} = [a_{ij}] \in \mathbb{R}^{n\times n}$, where $a_{ii} = 0$ and $a_{ij} \geq 0$ ($a_{ij} = a_{ji} > 0$ if and only if there is an edge between nodes j and i). The neighbor set of agent i is defined as $\mathcal{N}_i = \{j\colon (j, i) \in \mathcal{E}\}$ for $i = 1, 2, \cdots, n$. A path in graph $\mathcal{G}$ is an alternating sequence $i_1e_1i_2e_2 \cdots e_{k-1}i_k$ of nodes i_l and edges $e_m = (i_m, i_{m+1}) \in \mathcal{E}$ for $l = 1, 2, \cdots, k$. If there exists a path from node i to node j then node i is said to be reachable from node j. The Laplacian $L = [l_{ij}] \in \mathbb{R}^{n\times n}$ of graph $\mathcal{G}$ is defined as $l_{ii} = \sum_{j\neq i} a_{ij}$ and $l_{ij} = -a_{ij} (j \neq i)$. It can be found that the Laplacian is symmetric and semi-definite. Denote its ordered eigenvalues as $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N$. The corresponding eigenvector of $\lambda_1 = 0$ is the all one vector $\mathbf{1}_N$. Moreover, $\lambda_2 > 0$ if and only if this graph $\mathcal{G}$ is connected.

2.2 Convex Analysis

For each set $X \in \mathbb{R}^m$, the indicator function is denoted by δ_X with $\delta_X(x) = 0$ for any $x \in X$ and $\delta_X(x) = \infty$ for any $x \notin X$. A set $X \in \mathbb{R}^m$ is said to be convex if $\theta x + (1 - \theta)y \in X$ for any $x, y \in X$ and $\theta \in (0, 1)$. For a closed convex set $X \neq \emptyset$, the projection operator $P_X\colon \mathbb{R}^m \to X$ is defined as $P_X[x] = \arg\min_{y\in X} \|y - x\|$. The projection operator is non-expansive in the sense that $\|P_X[x] - P_X[y]\| \leq \|x - y\|$ for any $x, y \in \mathbb{R}^m$. For any $x \in \mathbb{R}^m$, it holds $(x - P_X[x])^{\mathrm{T}}(y - P_X[x]) \leq 0$, $\forall y \in X$.

For a given function $f\colon \mathbb{R}^m \to \mathbb{R}$, we denote by $\mathrm{dom}\, f = \{x \in \mathbb{R}^m \mid |f(x)| < \infty\}$ the domain of f. We always assume that $\mathrm{dom}\, f \neq \emptyset$ and $\mathrm{dom}\, f = \mathbb{R}^m$ if not specified. We say it is convex if its domain is convex and $f(\alpha x + (1 - \alpha y)) \leq \alpha f(x) + (1 - \alpha)f(y)$ holds for all $x, y \in \mathrm{dom}\, f$ and $\alpha \in [0, 1]$. If this inequality is strict in the sense that the equation holds only if $x = y$, the function is called strictly convex. A function f is called closed and convex on a convex set $X \subset \mathrm{dom}\, f$ if its constrained epigraph $\mathrm{epi}_X(f) = \{(x, t) \in X \times \mathbb{R} \mid t \geq f(x)\}$ is a closed convex set. If $X = \mathrm{dom}\, f$, we call f a closed convex function.

A vector-valued function $\boldsymbol{f}\colon \mathbb{R}^m \to \mathbb{R}^m$ is Lipschitz with constant $\vartheta > 0$ (or simply ϑ-Lipschitz) if we have

$$\|\boldsymbol{f}(\zeta_1) - \boldsymbol{f}(\zeta_2)\| \leq \vartheta\|\zeta_1 - \zeta_2\|, \quad \forall \zeta_1, \zeta_2 \in \mathbb{R}^m.$$

Let us consider a function $\phi\colon X \times Z \to \mathbb{R}$, where X and Z are nonempty subsets of $\mathbb{R}^n$ and $\mathbb{R}^m$, respectively. A pair of vectors $x^* \in X$ and $z^* \in Z$ is called a saddle point of ϕ if

$\phi(x^*, z) \le \phi(x^*, z^*) \le \phi(x, z^*)$ holds for any $x \in X$ and $z \in Z$.

3 Problem Formulation

In this paper, we focus on solving the following constrained optimization problem by a network of N agents:

$$\begin{aligned} \min \quad & f(x) = \sum_{i=1}^{N} f_i(x) \\ \text{s.t.} \quad & x \in X \triangleq \bigcap_{i=1}^{N} X_i. \end{aligned} \tag{1}$$

Here function $f_i\colon \mathbb{R} \to \mathbb{R}$ and set X_i are private to agent i for each $i \in \mathcal{N} \triangleq \{1, 2, \cdots, N\}$ and can not be shared with others.

To ensure its solvability, the following assumption is made.

Assumption 3.1 For each $i \in \mathcal{N}$, function $f_i\colon \mathbb{R} \to \mathbb{R}$ is convex, set X_i is convex and closed, and $\operatorname{int} X = \bigcap_{i\in\mathcal{N}} \operatorname{int} X_i$ is nonempty and contained in $\operatorname{dom} f_i$.

Note that f_i might be nondifferentiable under this assumption. Denote the minimal value of the problem (1) by f^* and the optimal solution set by $\mathcal{X}^*$, i.e., $f^* = \min_{x\in X} f(x)$ and $\mathcal{X}^* = \{x \in X \,|\, f(x) = f^*\}$. As usual, we assume f^* is finite and set $\mathcal{X}^*$ is nonempty. To cooperatively address the optimization problem (1) in a distributed manner, we use a weighted undirected graph $\mathcal{G} = \{\mathcal{N}, \mathcal{E}, \mathcal{A}\}$ to describe the information sharing relationships with node set $\mathcal{N}$, edge set $\mathcal{E} \subset \mathcal{N} \times \mathcal{N}$, and weight matrix $\mathcal{A} = [a_{ij}]_{N\times N}$. Here $a_{ij} = a_{ji} > 0$ means agents i and j can communicate with each other.

Assumption 3.2 Graph $\mathcal{G}$ is connected.

Suppose that agent i maintains an estimate x_i of the optimal solution to (1) with other (possible) auxiliary variables. Agents exchange these variables through the communication network described by $\mathcal{G}$ and perform some updates at given discrete-time instants $k = 1, 2, \cdots$. Then, the distributed optimization problem in this paper is formulated to find an update rule of $x_i(k)$ for agent i using only its own and neighboring information such that $\lim_{k\to\infty}[x_i(k) - x_j(k)] = 0$ for any $i, j \in \mathcal{N}$ and $\lim_{k\to\infty} \sum_{i=1}^{N} f_i(x_i(k)) = f^*$. If possible, we expect that all the estimates will converge to an optimal solution to the problem (1).

As stated above, this problem has been intensively studied in literature [1, 2]. However, most existing designs require the exact subgradients of the local objective functions in constructing effective distributed algorithms. In this paper, we are interested in the solvability of the formulated distributed optimization problem (1) working with inexact subgradients of the local objective functions. For this purpose, we adopt the notion of ε-subgradient to describe such inexactness as in [4, 18] and assume the ε-subgradient of f_i can be easily computed for any given $\varepsilon \ge 0$. It should be noted that assuming the knowledge of ε-subgradient is more conceptual than concrete. In practice, ε-subgradient often arises in the numerical approximation of classical algorithms, e.g., proximal method and bundle method[3, 4, 16, 19].

Definition 3.1 For a convex function $f\colon \mathbb{R}^m \to \mathbb{R}$ and a scalar $\varepsilon > 0$, $g \in \mathbb{R}^m$ is said to be a ε-subgradient of f at $x \in \mathbb{R}^m$ if

$$f(y) \geq f(x) + g^{\mathrm{T}}(y - x) - \varepsilon, \quad \forall y \in \operatorname{dom} f.$$

Denote by $\partial_\varepsilon f(x)$ the ε-subdifferential of f at $x \in \mathbb{R}^m$, which is the set of all ε-subgradients of f at x. $\partial_\varepsilon f(x)$ is nonempty and convex for any $x \in \mathbb{R}^m$ due to the convexity of f. Moreover, $\partial_0 f(x)$ coincides with the subdifferential of f at $x \in \mathbb{R}^m$.

In next section, we will convert our problem into a saddle-point seeking problem and develop a projected primal-dual ε-subgradient method with rigorous solvability analysis.

4 Main Result

To begin with, we rewrite the problem (1) into an alternative form as in [11, 20]:

$$\begin{aligned} \min \quad & \widetilde{f}(\boldsymbol{x}) = \sum_{i=1}^{N} f_i(x_i) \\ \text{s.t.} \quad & L\boldsymbol{x} = \mathbf{0}_N, \\ & \boldsymbol{x} \in \widetilde{X} \triangleq X_1 \times \cdots \times X_N, \end{aligned} \tag{2}$$

where $\boldsymbol{x} = \operatorname{col}(x_1, \cdots, x_N)$ and L is the Laplacian of graph $\mathcal{G}$. Note that L is symmetric and positive semi-definite with its ordered eigenvalues as $0 = \lambda_1 < \lambda_2 \leq \cdots \lambda_N$ under Assumption 3.2 by Theorem 2.8 in [21].

Consider the augmented Lagrangian function of the problem (2):

$$\Phi(\boldsymbol{x}, \boldsymbol{v}) = \widetilde{f}(\boldsymbol{x}) + \boldsymbol{v}^{\mathrm{T}} L\boldsymbol{x} + \frac{1}{2}\boldsymbol{x}^{\mathrm{T}} L\boldsymbol{x} \tag{3}$$

with $\boldsymbol{v} = \operatorname{col}(v_1, \cdots, v_N) \in \mathbb{R}^N$. By Proposition 3.4.1 in [4], if Φ has a saddle point $(\boldsymbol{x}^*, \boldsymbol{v}^*)$ in $\widetilde{X} \times \mathbb{R}^N$, then $\boldsymbol{x}^*$ must be an optimal solution to the problem (2), which in turn provides an optimal solution to (1). Since the Slater's condition holds under Assumption 3.1, such saddle points indeed exist by virtue of Theorems 3.34 and 4.7 in [22]. Thus, it suffices for us to seek a saddle point of Φ in $\widetilde{X} \times \mathbb{R}^N$.

Following this conversion, many solvability results on the problem (1) have been presented when the exact gradient or subgradient of f_i is available, e.g., [11, 17, 20, 23–25]. However, whether and how ε-subgradient algorithms can be derived has not been discussed yet. To this end, we are motivated by aforementioned saddle-point seeking designs and present the following dynamics:

$$\begin{aligned} x_i(k+1) &= P_{X_i}[x_i(k) - \alpha_k(g_i(k) + \widehat{x}_i(k) + \widehat{v}_i(k))], \\ v_i(k+1) &= v_i(k) + \alpha_k \widehat{x}_i(k), \end{aligned} \tag{4}$$

where $\widehat{x}_i(k) \triangleq \sum_{j=1}^{N} a_{ij}(x_i(k) - x_j(k))$, $\widehat{v}_i(k) \triangleq \sum_{j=1}^{N} a_{ij}(v_i(k) - v_j(k))$, and $g_i(k) \in \partial_{\varepsilon_k} f_i(x_i(k))$ with parameters ε_k, $\alpha_k > 0$ to be specified later. It can be taken as a constrained version of

algorithms in [23, 24]. Different from similar primal-dual designs in [11, 20], we do not require the differentiability of these objective functions or their exact gradients.

Letting $\boldsymbol{x}(k) = \mathrm{col}(x_1(k), \cdots, x_N(k))$ and $\boldsymbol{v}(k) = \mathrm{col}(v_1(k), \cdots, v_N(k))$, we can put (4) into a compact form:

$$\begin{aligned} \boldsymbol{x}(k+1) &= P_{\widetilde{X}}[\boldsymbol{x}(k) - \alpha_k(\boldsymbol{g}(k) + L\boldsymbol{v}(k) + L\boldsymbol{x}(k))], \\ \boldsymbol{v}(k+1) &= \boldsymbol{v}(k) + \alpha_k L\boldsymbol{x}(k) \end{aligned} \tag{5}$$

with $\boldsymbol{g}(k) = \mathrm{col}(g_1(k), \cdots, g_N(k)) \in \partial_{N\varepsilon_k} \widetilde{f}(\boldsymbol{x}(k)) \in \mathbb{R}^N$. It can be further rewritten as follows:

$$\boldsymbol{z}(k+1) = P_{\overline{X}}[\boldsymbol{z}(k) - \alpha_k T_{\varepsilon_k}(\boldsymbol{z}(k))], \tag{6}$$

where $\boldsymbol{z}(k) = \mathrm{col}(\boldsymbol{x}(k), \boldsymbol{v}(k))$, $\overline{X} = \widetilde{X} \times \mathbb{R}^N$, and

$$T_{\varepsilon_k}(\boldsymbol{z}(k)) = \begin{bmatrix} \boldsymbol{g}(k) + L\boldsymbol{v}(k) + L\boldsymbol{x}(k) \\ -L\boldsymbol{x}(k) \end{bmatrix}.$$

To establish the effectiveness of this algorithm, another assumption is made as follows.

Assumption 4.1 The ε_k-subgradient sequence $\{g_i(k)\}$ is uniformly bounded for each i, i.e., there exists a scalar $C > 0$ such that $\max_{i\in\mathcal{N}}\{\|g_i(k)\|\} < C$ all $k > 0$.

This assumption is temporally made for simplicity as in [9, 26] and will be further removed later by some novel step sizes later. Suppose $\boldsymbol{z}^* = \mathrm{col}(\boldsymbol{x}^*, \boldsymbol{v}^*)$ is a saddle point of Φ in $\widetilde{X} \times \mathbb{R}^N$. Here is a key lemma under Assumption 4.1.

Lemma 4.1 *Suppose Assumptions* 3.1, 3.2, 4.1 *hold. Along the trajectory of the algorithm* (4)*, there exists some* $C_1 > 0$ *such that, for any* $k = 1, 2, \cdots$*, the following inequality holds:*

$$\|\boldsymbol{z}(k+1) - \boldsymbol{z}^*\|^2 \le (1 + C_1\alpha_k^2)\|\boldsymbol{z}(k) - \boldsymbol{z}^*\|^2 - 2\alpha_k\Delta(\boldsymbol{x}(k)) + 2N\alpha_k\varepsilon_k + C_1\alpha_k^2, \tag{7}$$

where $\Delta(\boldsymbol{x}(k)) \triangleq \Phi(\boldsymbol{x}(k), \boldsymbol{v}^*) - \Phi(\boldsymbol{x}^*, \boldsymbol{v}^*) + \frac{1}{2}\boldsymbol{x}(k)^{\mathrm{T}} L\boldsymbol{x}(k) \ge 0$.

Proof By lemma conditions, $(\boldsymbol{x}^*, \boldsymbol{v}^*)$ is a saddle point of Φ. Then, $\Delta(\boldsymbol{x}(k)) \triangleq \Phi(\boldsymbol{x}(k), \boldsymbol{v}^*) - \Phi(\boldsymbol{x}^*, \boldsymbol{v}^*) + \frac{1}{2}\boldsymbol{x}(k)^{\mathrm{T}} L\boldsymbol{x}(k) \ge 0$ can be easily verified by the definition of saddle points.

Next, we consider the evolution of $\|\boldsymbol{z}(k) - \boldsymbol{z}^*\|^2$ with respect to k. Under the iteration (4), it follows then

$$\begin{aligned} &\|\boldsymbol{z}(k+1) - \boldsymbol{z}^*\|^2 \\ &= \|P_{\overline{X}}[\boldsymbol{z}(k) - \alpha_k T_{\varepsilon_k}(\boldsymbol{z}(k))] - \boldsymbol{z}^*\|^2 \\ &\le \|\boldsymbol{z}(k) - \alpha_k T_{\varepsilon_k}(\boldsymbol{z}(k)) - \boldsymbol{z}^*\|^2 - \|\boldsymbol{z}(k) - \alpha_k T_{\varepsilon_k}(\boldsymbol{z}(k)) - P_{\overline{X}}[\boldsymbol{z}(k) - \alpha_k T_{\varepsilon_k}(\boldsymbol{z}(k))]\|^2 \\ &\le \|\boldsymbol{z}(k) - \boldsymbol{z}^*\|^2 - 2\alpha_k(\boldsymbol{z}(k) - \boldsymbol{z}^*)^{\mathrm{T}} T_{\varepsilon_k}(\boldsymbol{z}(k)) + \alpha_k^2\|T_{\varepsilon_k}(\boldsymbol{z}(k))\|^2. \end{aligned} \tag{8}$$

By the proprieties of saddle point and ε_k-subgradient, we have

$$(\boldsymbol{z}(k) - \boldsymbol{z}^*)^{\mathrm{T}} T_{\varepsilon_k}(\boldsymbol{z}(k)) = (\boldsymbol{x}(k) - \boldsymbol{x}^*)^{\mathrm{T}}(\boldsymbol{g}(k) + L\boldsymbol{v}(k) + L\boldsymbol{x}(k)) - (\boldsymbol{v}(k) - \boldsymbol{v}^*)^{\mathrm{T}} L\boldsymbol{x}(k)$$

$$\begin{aligned}
&\geq \widetilde{f}(\boldsymbol{x}(k)) - \widetilde{f}(\boldsymbol{x}^*) - N\varepsilon_k + {\boldsymbol{v}^*}^{\mathrm{T}} L\boldsymbol{x}(k) + \boldsymbol{x}(k)^{\mathrm{T}} L\boldsymbol{x}(k) \\
&= \Phi(\boldsymbol{x}(k), \boldsymbol{v}^*) - \Phi(\boldsymbol{x}^*, \boldsymbol{v}^*) + \frac{1}{2}\boldsymbol{x}(k)^{\mathrm{T}} L\boldsymbol{x}(k) - N\varepsilon_k \\
&= \Delta(\boldsymbol{x}(k)) - N\varepsilon_k. \qquad (9)
\end{aligned}$$

Since $L\boldsymbol{x}^* = \boldsymbol{0}$, $T_{\varepsilon_k}(\boldsymbol{z}(k))$ can be rewritten as

$$T_{\varepsilon_k}(\boldsymbol{z}(k)) = \begin{bmatrix} \boldsymbol{g}(k) + L(\boldsymbol{x}(k) - \boldsymbol{x}^*) + L(\boldsymbol{v}(k) - \boldsymbol{v}^*) + L\boldsymbol{v}^* \\ -L(\boldsymbol{x}(k) - \boldsymbol{x}^*) \end{bmatrix}.$$

Under Assumption 4.1, there must be constant $C_1 > 0$ such that

$$\|T_{\varepsilon_k}(\boldsymbol{z}(k))\|^2 \leq C_1(1 + \|\boldsymbol{z}(k) - \boldsymbol{z}^*\|^2). \qquad (10)$$

Putting all inequalities (8)–(10) together, we have

$$\|\boldsymbol{z}(k+1) - \boldsymbol{z}^*\|^2 \leq (1 + C_1\alpha_k^2)\|\boldsymbol{z}(k) - \boldsymbol{z}^*\|^2 - 2\alpha_k\Delta(\boldsymbol{x}(k)) + 2N\alpha_k\varepsilon_k + C_1\alpha_k^2,$$

which is exactly the expected inequality (7). ∎

When the exact subgradient is available (i.e., $\varepsilon_k = 0$), the inequality (7) can be simplified into the well-known supermartingale inequality ensuring the convergence of $\boldsymbol{z}(k)$ towards $\boldsymbol{z}^*$ as shown in [4] if $\{\alpha_k\}$ is chosen to satisfy

$$\sum_{k=1}^{\infty} \alpha_k = \infty, \quad \sum_{k=1}^{\infty} \alpha_k^2 < \infty. \qquad (11)$$

However, the inexactness of available subgradients deteriorates this property and the expected convergence might fail when we use only ε-subgradients in the iteration (4).

Let us denote $\overline{\Delta} = \liminf_{k\to\infty} \Delta(\boldsymbol{x}(k))$ and take a closer look at the inequality (7). Note that $\overline{\Delta}$ consists of two parts, i.e., the discrepancy of function values and violation of constraints (in term of consensus error) of the iterative sequence. In fact, when $\overline{\Delta} = 0$, the associated iterative sequences necessarily reach a consensus such that $\lim_{k\to\infty} \widetilde{f}(\boldsymbol{x}(k)) = f^*$. Thus, this value $\overline{\Delta}$ can be taken as a measure of suboptimality of the iterative sequence. In other words, we can determine the upper bound for $\overline{\Delta}$ to evaluate the effectiveness of the algorithm (4).

We first consider the case when ε_k is constant.

Theorem 4.2 *Suppose Assumptions* 3.1, 3.2, 4.1 *hold. Let the step size α_k be chosen to satisfy* (11) *and $\varepsilon_k = \varepsilon_0 > 0$ for all $k > 0$. Then, along the trajectory of the algorithm* (4)*, it holds that*

$$0 \leq \overline{\Delta} \leq N\varepsilon_0. \qquad (12)$$

Proof To prove this theorem, we only have to show that $\overline{\Delta} \leq N\varepsilon_0$. If the inequality does not hold, there must exist a $\delta > 0$ and a sufficient large integer $K_1 > 1$ such that $\Delta(\boldsymbol{x}(k)) > N\varepsilon_0 + \delta$ for all $k \geq K_1$. By (11), we have $\lim_{k\to\infty} \alpha_k = 0$. Thus, there must exist an integer $K_2 > 1$

such that $0 < \alpha_k \leq \frac{\delta}{C_1}$ for all $k \geq K_2$. Bringing these conditions together, one can strengthen the inequality (7) for all $k \geq K \triangleq \max\{K_1, K_2\}$ as follows:

$$\|\boldsymbol{z}(k+1) - \boldsymbol{z}^*\|^2 \leq (1 + C_1\alpha_k^2)\|\boldsymbol{z}(k) - \boldsymbol{z}^*\|^2 - \alpha_k\delta.$$

Summing up its both sides from K to $\overline{K} > K$ gives

$$\|\boldsymbol{z}(\overline{K}+1) - \boldsymbol{z}^*\|^2 \leq \|\boldsymbol{z}(K) - \boldsymbol{z}^*\|^2 \prod_{k=K}^{\overline{K}} (1 + C_1\alpha_k^2) - \delta \sum_{k=K}^{\overline{K}} \alpha_k,$$

where we use $1 + C_1\alpha_k^2 > 1$ to handle the cross terms.

Note that $1 + \theta \leq \mathrm{e}^{\theta}$ for any $\theta > 0$. Hence $\prod_{k=K}^{\overline{K}}(1 + C_1\alpha_k^2) \leq \mathrm{e}^{C_1 \sum_{k=K}^{\overline{K}} \alpha_k^2} \leq \mathrm{e}^{C_1 \sum_{k=1}^{\infty} \alpha_k^2}$. Under the condition (11), there must exist a positive scalar $\overline{C} > 0$ such that

$$\|\boldsymbol{z}(\overline{K}+1) - \boldsymbol{z}^*\|^2 \leq \overline{C}\|\boldsymbol{z}(K) - \boldsymbol{z}^*\|^2 - \delta \sum_{k=K}^{\overline{K}} \alpha_k,$$

which can not hold for a sufficiently large $\overline{K}$ since $\sum_{k=1}^{\infty} \alpha_k = \infty$. We obtain a contradiction and complete the proof. ∎

Remark 4.3 According to Theorem 4.2, one can generally obtain a suboptimal solution to the problem (1) using inexact subgradients. If we are interested in an exact solution, it is required to ensure $\lim_{k\to\infty} \varepsilon_k = 0$. For a very special case when $\varepsilon_k = 0$, it shows the effectiveness of our algorithm (4) in solving the formulated problem (1) with exact subgradients. This observation is consistent with the existing subgradient methods in [12–14, 17, 24].

With Theorem 4.2, it is natural for us to enforce some stronger condition on the error ε_k for a better convergence performance of the entire sequence $\{x_i(k)\}$. Along this line, we provide another theorem supposing the accumulated error of subgradient inexactness is not too large.

Theorem 4.4 *Suppose Assumptions* 3.1, 3.2, 4.1 *hold. Let the parameters* α_k, $\varepsilon_k > 0$ *be chosen to satisfy the following condition:*

$$\sum_{k=1}^{\infty} \alpha_k = \infty, \quad \sum_{k=1}^{\infty} \alpha_k^2 < \infty, \quad \sum_{k=1}^{\infty} \alpha_k\varepsilon_k < \infty. \tag{13}$$

Then, along the trajectory of the algorithm (4)*, we have:*

1) *The sequence* $\{\|\boldsymbol{z}(k+1) - \boldsymbol{z}^*\|\}$ *converges;*

2) *The estimates* $x_1(k), \cdots, x_N(k)$ *reach an optimal consensus in the sense that* $\lim_{k\to\infty} [x_i(k) - x_j(k)] = 0$ *and* $\lim_{k\to\infty} \widetilde{f}(\boldsymbol{x}(k)) = \widetilde{f}(\boldsymbol{x}^*) = f^*$*;*

3) $\{\boldsymbol{z}(k)\}$ *has at least one cluster point* $\overline{\boldsymbol{z}} = \mathrm{col}(\overline{\boldsymbol{x}}, \overline{\boldsymbol{v}})$ *such that* $\overline{\boldsymbol{x}} = \mathbf{1}_N x^*$ *with* x^* *being an optimal solution to the problem* (1)*;*

4) *If the optimal solution to the problem* (1) *is unique, i.e.,* $\mathcal{X}^* = \{x^*\}$*, then* $\lim_{k\to\infty} x_i(k) = x^*$ *for each* $i \in \mathcal{N}$*.*

Proof Note that $\Delta(\boldsymbol{x}(k)) \geq 0$ by Lemma 4.1 and $\sum_{k=1}^{\infty} \alpha_k\varepsilon_k + \sum_{k=1}^{\infty} \alpha_k^2 < \infty$ under the theorem assumption. Applying Lemma 5.31 in [27] to the inequality (7), we can obtain the

convergence of $\{\|\boldsymbol{z}(k+1)-\boldsymbol{z}^*\|\}$ and

$$0 \leq \sum_{i=1}^{\infty} \alpha_k \Delta(\boldsymbol{x}(k)) < \infty. \tag{14}$$

Thus, the sequence $\{\boldsymbol{z}(k)\}$ must be uniformly bounded by some $C_2 > 0$. From the continuity of $\Delta(\boldsymbol{x})$, it must be C_3-Lipschitz with respect to $\boldsymbol{x}$ for some constant $C_3 > 0$. It follows then

$$\begin{aligned}
\Delta(\boldsymbol{x}(k+1)) - \Delta(\boldsymbol{x}(k)) &\leq C_3 \|\boldsymbol{x}(k+1) - \boldsymbol{x}(k)\| \\
&= C_3 \|P_{\widetilde{X}}[\boldsymbol{x}(k) - \alpha_k(\boldsymbol{g}(k) + L\boldsymbol{v}(k) + L\boldsymbol{x}(k))] - \boldsymbol{x}(k)\| \\
&\leq C_3 \|\boldsymbol{x}(k) - \alpha_k(\boldsymbol{g}(k) + L\boldsymbol{v}(k) + L\boldsymbol{x}(k)) - \boldsymbol{x}(k)\| \\
&\leq C_3 \alpha_k \|\boldsymbol{g}(k) + L\boldsymbol{v}(k) + L\boldsymbol{x}(k)\| \\
&\leq C_3(\sqrt{N}C + 2\lambda_{\max}(L)C_2)\alpha_k.
\end{aligned} \tag{15}$$

Jointly using (14), (15), and $\sum_{k=1}^{\infty} \alpha_k = \infty$, we resort to Proposition 2 in [15] and conclude that $\lim_{k\to\infty} \Delta(\boldsymbol{x}(k)) = 0$. Recalling the expression of Δ, we have $\lim_{k\to\infty} \boldsymbol{x}^{\mathrm{T}}(k) L \boldsymbol{x}(k) = 0$ and $\lim_{k\to\infty}[\widetilde{f}(\boldsymbol{x}(k)) + \boldsymbol{v}^{*\mathrm{T}} L \boldsymbol{x}(k)] = \lim_{k\to\infty} \widetilde{f}(\boldsymbol{x}(k)) = \widetilde{f}(\boldsymbol{x}^*)$. Note that L is positive semidefinite with 0 as its simple eigenvalue under Assumption 3.2. It follows then $\lim_{k\to\infty}[x_i(k) - x_j(k)] = 0$ and $\lim_{k\to\infty} \widetilde{f}(\boldsymbol{x}(k)) = \widetilde{f}(\boldsymbol{x}^*) = f^*$.

Due to the uniform boundedness of sequence $\{\boldsymbol{z}(k)\}$ by item 1), there must be a convergent subsequence $\{\boldsymbol{z}_{k_m}\}$ of $\{\boldsymbol{z}_k\}$. We denote its limit by $\overline{\boldsymbol{z}} = \mathrm{col}(\overline{\boldsymbol{x}}, \overline{\boldsymbol{v}})$. Then, it should satisfy $L\overline{\boldsymbol{x}} = 0$ and $\widetilde{f}(\overline{\boldsymbol{x}}) = \lim_{m\to\infty} \widetilde{f}(\boldsymbol{x}_{k_m}) = f^*$ by item 2). In other words, $\overline{\boldsymbol{x}}$ is an optimal solution to (2). By Assumption 3.2, one can conclude that there exists some $\overline{x} \in \mathbb{R}$ such that $\overline{\boldsymbol{x}} = \mathbf{1}_N \overline{x}$. Note that $f(\overline{x}) = \widetilde{f}(\overline{\boldsymbol{x}}) = f^*$, i.e., $\overline{x}$ is an exact optimal solution to the problem (1).

If $\mathcal{X}^* = \{x^*\}$ holds, all convergent subsequences of $\{\boldsymbol{x}(k)\}$ have the same limit x^*. This combined with the boundedness of $\{\boldsymbol{z}(k)\}$ implies item 4) and completes the proof. ∎

Remark 4.5 This theorem specifies a nontrivial case when our distributed optimization problem (1) can be exactly solved even using only inexact subgradients information of the local objective functions. This observation is consistent with the centralized results in [6]. Compared with similar primal-dual domain results in [11, 17, 20, 23, 25, 28], this algorithm further allows us to consider nonsmooth objective functions with only approximate subgradients.

It is known that normalization might improve the transient performance of the algorithms to avoid overshoots at the starting phase. However, conventional normalized techniques often involve some global information and can not be directly implemented in distributed settings. We here present a novel componentwise normalized version of the algorithm (4) as follows:

$$\begin{aligned}
x_i(k+1) &= P_{X_i}\left[x_i(k) - \frac{\alpha_k}{\max\{c,\, \delta_{ik,D}\}}(g_i(k) + \widehat{x}_i(k) + \widehat{v}_i(k))\right], \\
v_i(k+1) &= v_i(k) + \frac{\alpha_k}{\max\{c,\, \delta_{ik,D}\}}\widehat{x}_i(k), \\
\delta_{ik,\,m} &= \begin{cases} \|T^i_{\varepsilon_k}(\boldsymbol{z}(k))\|, & \text{when } m = 1, \\ \max\limits_{j\in\mathcal{N}_i}\{\delta_{ik,\,m-1},\, \delta_{jk,\,m-1}\}, & \text{when } 2 \leq m \leq D, \end{cases}
\end{aligned} \tag{16}$$

where integer $D \geq \mathrm{D}(\mathcal{G}) + 1$ with $\mathrm{D}(\mathcal{G})$ the diameter of graph $\mathcal{G}$ and $c > 0$ is any given constant. Since $\mathrm{D}(\mathcal{G})$ or an upper bound can be computed by distributed rules[29], this normalized algorithm is implementable in a fully distributed manner by embedding a max-consensus subiteration.

Here is a corollary to state that the normalized algorithm (16) retains all the established properties in Theorem 4.4.

Corollary 4.6 *Suppose Assumptions* 3.1–3.2 *hold. Choose the same parameters satisfying condition* (13). *Then, along the trajectory of the algorithm* (16), *the sequence* $\{\boldsymbol{z}(k)\}$ *retains all the established properties in Theorem* 4.4.

1) *The sequence* $\{\|\boldsymbol{z}(k+1) - \boldsymbol{z}^*\|\}$ *converges;*

2) *The estimates* $x_1(k), \cdots, x_N(k)$ *reach an optimal consensus in the sense that* $\lim_{k\to\infty} [\boldsymbol{x}_i(k) - \boldsymbol{x}_j(k)] = 0$ *and* $\lim_{k\to\infty} \widetilde{f}(\boldsymbol{x}(k)) = \widetilde{f}(\boldsymbol{x}^*) = f^*$;

3) $\{\boldsymbol{z}(k)\}$ *has at least one cluster point* $\overline{\boldsymbol{z}} = \mathrm{col}(\overline{\boldsymbol{x}}, \overline{\boldsymbol{v}})$ *such that* $\overline{\boldsymbol{x}} = \mathbf{1}x^*$ *with* x^* *being an optimal solution to the problem* (1).

4) *If the optimal solution to the problem* (1) *is unique, i.e.,* $\mathcal{X}^* = \{x^*\}$, *then* $\lim_{k\to\infty} x_i(k) = x^*$ *for each* $i \in \mathcal{N}$.

Proof The proof is similar with that of Theorem 4.4. First, we recall Theorem 4.1 in [30] and conclude that all agents will get $\max\{c, \max_{i\in\mathcal{N}} \|T^i_{\varepsilon_k}(\boldsymbol{z}(k))\|\}$ after the subiteration. Thus, we only have to consider the following system:

$$\begin{aligned} x_i(k+1) &= P_{X_i}[x_i(k) - \alpha_k\gamma_k(g_i(k) + \widehat{x}_i(k) + \widehat{v}_i(k))], \\ v_i(k+1) &= v_i(k) + \alpha_k\gamma_k\widehat{x}_i(k) \end{aligned}$$

with $\gamma_k = \frac{1}{\max\{c, \max_{i\in\mathcal{N}} \|T^i_{\varepsilon_k}(\boldsymbol{z}(k))\|\}}$. For this new system, we can establish a similar inequality as (7) for $\|\boldsymbol{z}(k) - \boldsymbol{z}^*\|$:

$$\|\boldsymbol{z}(k+1) - \boldsymbol{z}^*\|^2 \leq \|\boldsymbol{z}(k) - \boldsymbol{z}^*\|^2 - 2\alpha_k\gamma_k(\boldsymbol{z}(k) - \boldsymbol{z}^*)^{\mathrm{T}}T_{\varepsilon_k}(\boldsymbol{z}(k)) + \alpha_k^2\gamma_k^2\|T_{\varepsilon_k}(\boldsymbol{z}(k))\|^2.$$

Note that $\gamma_k^2\|T_{\varepsilon_k}(\boldsymbol{z}(k))\|^2 \leq N$ and $0 \leq \gamma_k \leq \frac{1}{c}$. Recalling the fact (9) and $\Delta(\boldsymbol{x}(k)) \geq 0$, one can obtain

$$\|\boldsymbol{z}(k+1) - \boldsymbol{z}^*\|^2 \leq \|\boldsymbol{z}(k) - \boldsymbol{z}^*\|^2 - 2\alpha_k\gamma_k\Delta(\boldsymbol{x}(k)) + \frac{2N}{c}\alpha_k\varepsilon_k + N\alpha_k^2.$$

According to Lemma 5.31 in [27], we can conclude the convergence of $\{\|\boldsymbol{z}(k+1) - \boldsymbol{z}^*\|\}$ and $\sum_{k=1}^{\infty} \alpha_k\gamma_k\Delta(\boldsymbol{x}(k)) < \infty$. Then, $\{\|\boldsymbol{z}(k)\|\}$ is uniformly bounded, which implies that there must be small enough constant $C_4 > 0$ such that $\gamma_k \geq C_4 > 0$. We use Proposition 2 in [15] again and obtain that $\lim_{k\to\infty} \Delta(\boldsymbol{x}(k)) = 0$. Then, items 3) and 4) can be easily verified following a similar procedure as in Theorem 4.4. The proof is thus completed. ∎

Remark 4.7 Compared with the conventional normalized step sizes in [22, 31], the proposed componentwise normalized step size can be taken as their distributed extension and the iterative sequence generated by (16) might have a better transient behavior than that generated by (4). Interestingly, the widely used subgradient boundedness assumption (i.e., Assumption 4.1) is also removed as a byproduct, which might be favorable in distributed scenarios.

5 Simulation

In this section, we consider an LASSO (least absolute shrinkage and selection operator) regression problem to verify the effectiveness of our algorithms:

$$\min_{x\in X} f(x) = \frac{1}{2}\sum_{i=1}^{N} \|x - p_i\|^2 + \lambda N\|x\|_1,$$

where $\lambda N > 0$ is the regularization parameter, p_i is an estimate only known by agent i, and X is the constrained set. Letting $f_i(x) = \frac{1}{2}\|x - p_i\|^2 + \lambda\|x\|_1$ and $X_i = X$, we can put it into a form of the problem (1). Distributed subgradient algorithms have been developed to solve this problem when ∂f_i is available, e.g., [13]. Although ∂f_i and $\partial_\varepsilon f_i$ can be easily calculated, we use this example to conceptually show the effectiveness of our algorithm only using its ε-subgradient instead of the exact one. In fact, according to the definition of ε-subdifferential, we can obtain

$$\partial_\varepsilon f_i(x) = \begin{cases} \left[x - p_i - \lambda, x - p_i - \lambda - \dfrac{\lambda\varepsilon}{x}\right], & \text{for } x < -\dfrac{\varepsilon}{2}, \\ [x - p_i - \lambda, x - p_i + \lambda], & \text{for } x \in \left[-\dfrac{\varepsilon}{2}, \dfrac{\varepsilon}{2}\right], \\ \left[x - p_i + \lambda - \dfrac{\lambda\varepsilon}{x}, x - p_i + \lambda\right], & \text{for } x > \dfrac{\varepsilon}{2}. \end{cases}$$

For simulations, we set $N = 10$, $p_i = 2i$, and $\lambda = 0.1$. The communication graph is given as Figure 1 with unity weights. We assume this problem has heterogeneous local constraints specified by $X_i = [-20 + i, 15 - i]$ with $i = 1, 2, \cdots, 10$. In this case, $X \triangleq \bigcap_{i=1}^{N} X_i = [-10, 5]$ and the constrained minimizer to our problem is $x^* = 5$. Then, Assumptions 3.1–3.2 can be verified. In the simulations, we choose $x - p_i - \lambda - \frac{\lambda\varepsilon}{x}$, $x - p_i + \lambda$, and $x - p_i + \lambda - \frac{\lambda\varepsilon}{x}$ as the ε-subgradient at x for the three different conditions $x < -\frac{\varepsilon}{2}$, $-\frac{\varepsilon}{2} \le x \le \frac{\varepsilon}{2}$, and $x > \frac{\varepsilon}{2}$. To ensure the condition (13), we choose $\alpha_k = \varepsilon_k = \frac{3}{k+1}$. According to Theorem 4.4 and Corollary 4.6, the above problem can be solved by our proposed distributed primal-dual ε-subgradient method (PDεSM) (4) and the normalized primal-dual ε-subgradient method (NPDεSM) (16).

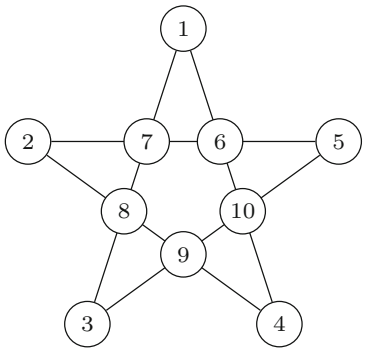

Figure 1 Communication graph $\mathcal{G}$ in our example

Simulation results with $\boldsymbol{x}(1) = \mathrm{col}(1,\ 0,\ 5,\ -1,\ 3,\ 2,\ 6,\ -2,\ -3,\ -4)$ and $c = 0.1$ for (4) and (16) are shown in Figures 2–3, where all agents' primal variables are observed to converge to the global optimal solution $x^* = 5$ while the dual variables are bounded and converge to some finite constants. This verifies the effectiveness of our algorithms. Comparing the performance of the two algorithms, one can find that although the normalized step size in (16) might slow down the convergence speed, the resultant transient performance of the primal and dual variables has been much improved with less and weaker oscillations. For a clear comparison, we let $e(k) = \frac{\|\boldsymbol{x}(k)-\mathbf{1}_{10}x^*\|}{\|\boldsymbol{x}(1)-\mathbf{1}_{10}x^*\|}$ be the residential error of our algorithms. The profiles of $e(k)$ in both algorithms are shown in Figure 4. From this, we can also confirm the improvement of transient performance by the proposed componentwise normalized step size.

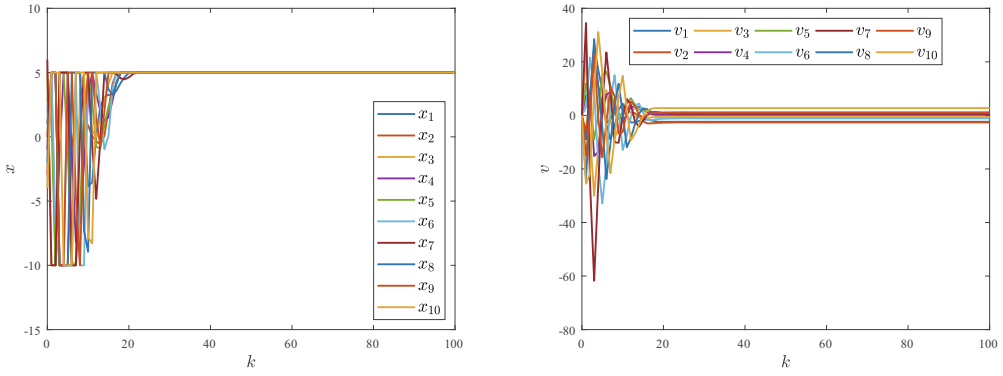

Figure 2 Profiles of primal and dual variables in Algorithm (4)

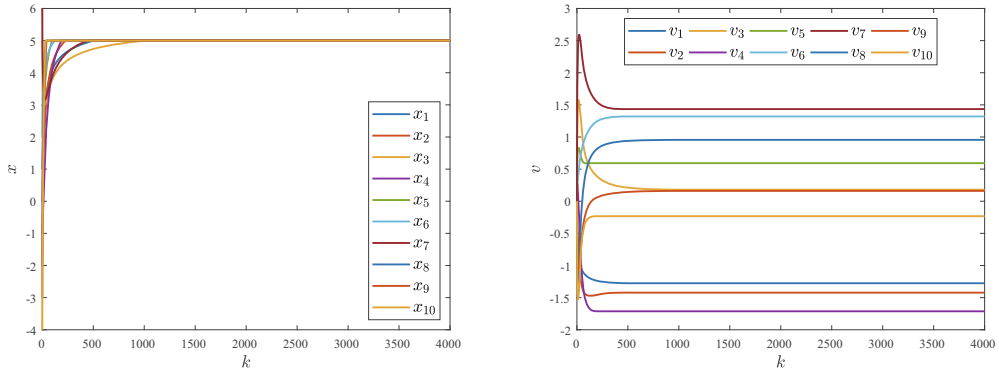

Figure 3 Profiles of primal and dual variables in Algorithm (16)

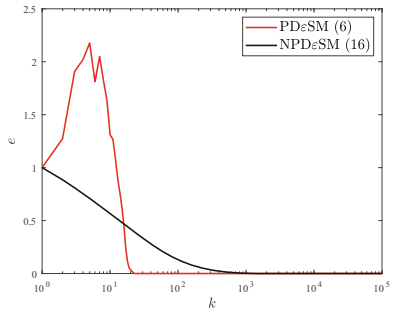

Figure 4 Profiles of residential errors in Algorithms (4) and (16)

6 Conclusion

In this paper, we have attempted to solve a distributed constrained optimization problem with inexact subgradient information of local objective functions. We have developed a projected primal-dual dynamics using only ε-subgradients and discussed its convergence properties. In particular, we have shown the exact solvability of this problem if the accumulated error introduced by subgradient inexactness is not too large. We have also presented a novel distributed normalized step size to improve the transient performance of our algorithms.

Note that the presented method is a conceptual framework to handle the inexact subgradient issue. It may be difficult to be implemented except that the objective functions have certain amenable structure, for example quadratic or piecewise linear. How to develop implementable and efficient forms of the method for more general functions deserves further investigation. Another interesting topic is to extend this work to more general communication graphs.

References

[1] Nedić A and Liu J, Distributed optimization for control, *Annual Review of Control, Robotics, and Autonomous Systems*, 2018, **1**: 77–103.

[2] Yang T, Yi X, Wu J, et al., A survey of distributed optimization, *Annual Reviews in Control*, 2019, **47**: 278–305.

[3] Correa R and Lemarchal C, Convergence of some algorithms for convex minimization, *Mathematical Programming*, 1993, **62**(1): 261–75.

[4] Bertsekas D, *Convex Optimization Algorithms*, Athena Scientific, Belmont, 2015.

[5] Devolder O, Glineur F, and Nesterov Y, First-order methods of smooth convex optimization with inexact oracle, *Mathematical Programming*, 2014, **146**(1–2): 37–75.

[6] Kiwiel K, Convergence of approximate and incremental subgradient methods for convex optimization, *SIAM Journal on Optimization*, 2004, **14**(3): 807–840.

[7] Jakovetić D, Bajović D, Xavier J, et al., Primal-dual methods for large-scale and distributed convex optimization and data analytics, *Proceedings of the IEEE*, 2020, **108**(11): 1923–1938.

[8] Nedić A and Ozdaglar A, Distributed subgradient methods for multi-agent optimization, *IEEE Transactions on Automatic Control*, 2009, **54**(1): 48–61.

[9] Jakovetić D, Moura J, and Xavier J, Linear convergence rate of a class of distributed augmented Lagrangian algorithms, *IEEE Transactions on Automatic Control*, 2014, **60**(4): 922–936.

[10] Yi P, Hong Y, and Liu F, Distributed gradient algorithm for constrained optimization with application to load sharing in power systems, *Systems & Control Letters*, 2015, **83**: 45–52.

[11] Lei J, Chen H, and Fang H, Primal-dual algorithm for distributed constrained optimization, *Systems & Control Letters*, 2016, **96**: 110–117.

[12] Xi C and Khan U, Distributed subgradient projection algorithm over directed graphs, *IEEE Transactions on Automatic Control*, 2016, **62**(8): 3986–3992.

[13] Liu S, Qiu Z, and Xie L, Convergence rate analysis of distributed optimization with projected subgradient algorithm, *Automatica*, 2017, **83**: 162–169.

[14] Zeng X, Yi P, and Hong Y, Distributed continuous-time algorithm for constrained convex optimizations via nonsmooth analysis approach, *IEEE Transactions on Automatic Control*, 2017, **62**(10): 5227–5233.

[15] Alber Y, Iusem A, and Solodov M, On the projected subgradient method for nonsmooth convex optimization in a Hilbert space, *Mathematical Programming*, 1998, **81**(1): 23–35.

[16] Auslender A and Teboulle M, Interior gradient and epsilon-subgradient descent methods for constrained convex minimization, *Mathematics of Operations Research*, 2004, **29**(1): 1–26.

[17] Zhu K, Zhu H, and Tang Y, On the boundedness of subgradients in distributed optimization, *Proceedings of* 39*th Chinese Control Conference* (*CCC*), Shenyang, 2020, 4912–4917.

[18] Polyak B, *Introduction to Optimization*, Optimization Software Inc., New York, 1987.

[19] Milln R and Machado M, Inexact proximal ε-subgradient methods for composite convex optimization problems, *Journal of Global Optimization*, 2019, **75**(4): 1029–1060.

[20] Liu Q, Yang S, and Hong Y, Constrained consensus algorithms with fixed step size for distributed convex optimization over multiagent networks, *IEEE Transactions on Automatic Control*, 2017, **62**(8): 4259–4265.

[21] Mesbahi M and Egerstedt M, *Graph Theoretic Methods in Multiagent Networks*, Princeton University Press, Princeton, 2010.

[22] Ruszczynski A, *Nonlinear Optimization*, Princeton University Press, Princeton, 2011.

[23] Wang J and Elia N, Control approach to distributed optimization, *Proceedings of* 48*th Annual Allerton Conference on Communication, Control, and Computing*, Monticello, 2010, 557–561.

[24] Gharesifard B and Cortés J, Distributed continuous-time convex optimization on weight-balanced digraphs, *IEEE Transactions on Automatic Control*, 2014, **59**(3): 781–786.

[25] Kia S, Cortés J, and Martínez S, Distributed convex optimization via continuous-time coordination algorithms with discrete-time communication, *Automatica*, 2015, **55**: 254–264.

[26] Nedić A and Ozdaglar A, Subgradient methods for saddle-point problems, *Journal of Optimization Theory & Applications*, 2009, **142**(1): 205–228.

[27] Bauschke H and Combettes P, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces* (2*nd Ed.*), Springer, Cham, 2017.

[28] Jakovetić D, Xavier J, and Moura J, Fast distributed gradient methods, *IEEE Transactions on Automatic Control*, 2014, **59**(5): 1131–1146.

[29] Oliva G, Setola R, and Hadjicostis C, Distributed finite-time average-consensus with limited computational and storage capability, *IEEE Transactions on Control of Network Systems*, 2016, **4**(2): 380–391.

[30] Nejad B, Attia S, and Raisch J, Max-consensus in a max-plus algebraic setting: The case of fixed communication topologies, Proceedings of 2009 *XXII International Symposium on Information, Communication and Automation Technologies*, Sarajevo, 2009, 1–7.

[31] Boyd S, Mutapcic A. Subgradient methods, *Notes for EE*364*b*, Stanford University, 2008.