Linear Convergence of First- and Zeroth-Order Primal–Dual Algorithms for Distributed Nonconvex Optimization

Xinlei Yi, Shengjun Zhang, Tao Yang, Tianyou Chai, and Karl H. Johansson

***Abstract*—This article considers the distributed nonconvex optimization problem of minimizing a global cost function formed by a sum of local cost functions by using local information exchange. We first consider a distributed first-order primal–dual algorithm. We show that it converges sublinearly to a stationary point if each local cost function is smooth and linearly to a global optimum under an additional condition that the global cost function satisfies the Polyak–Łojasiewicz condition. This condition is weaker than strong convexity, which is a standard condition for proving linear convergence of distributed optimization algorithms, and the global minimizer is not necessarily unique. Motivated by the situations where the gradients are unavailable, we then propose a distributed zeroth-order algorithm, derived from the considered first-order algorithm by using a deterministic gradient estimator, and show that it has the same convergence properties as the considered first-order algorithm under the same conditions. The theoretical results are illustrated by numerical simulations.**

***Index Terms*—Distributed nonconvex optimization, first-order algorithm, linear convergence, primal–dual algorithm, zeroth-order algorithm.**

I. Introduction

Distributed convex optimization has a long history, which can be traced back at least to the 1980s [1]. It has gained renewed interests in recent years due to its wide applications in power systems, machine learning, and sensor networks, just to name a few [2]. Various distributed optimization algorithms have been developed. Basic convergence results in distributed convex optimization typically ensure that algorithms converge to optimal points sublinearly (see, e.g., [3] and [4]). Linear convergence rate can be established under more stringent strong convexity conditions. For example, in [5]–[14] and [15]–[17], the authors assumed that each local cost function and the global cost function are strongly convex, respectively.

Unfortunately, in many practical applications, such as least squares, the cost functions are not strongly convex [18]. This situation has motivated researchers to consider alternatives to strong convexity. There are some results in centralized optimization. For instance, in [19], Necoara *et al.* derived linear convergence of several centralized first-order methods for smooth and constrained optimization problems when cost functions are convex and satisfy the quadratic functional growth condition; in [20], Karimi *et al.* showed linear convergence of centralized proximal-gradient methods for smooth optimization problems when cost functions satisfy the Polyak–Łojasiewicz (P–Ł) condition, which is weaker than the conditions assumed in [19]. There also are some results in distributed optimization [21]–[26]. Specifically, in [21], Shi *et al.* proposed the distributed exact first-order algorithm (EXTRA) to solve smooth convex optimization problems and proved linear convergence under the conditions that the global cost function is restricted strongly convex and the optimal set is a singleton, which are stronger than the P–Ł condition. The authors of [22] and [23] later extended the results in [21] to directed graphs. In [24], Yi *et al.* proposed a continuous-time distributed heavy-ball algorithm with event-triggered communication to solve smooth convex optimization problems and proved exponential convergence under the same conditions as that assumed in [21]. In [25], Liang *et al.* established linear convergence of the distributed primal–dual gradient descent algorithm for solving smooth convex optimization problems under the condition that the primal–dual gradient map is metrically subregular, which is different from the P–Ł condition but weaker than strong convexity. In [26], Yi *et al.* considered a distributed primal–dual gradient descent algorithm to solve smooth convex optimization problems and established linear convergence under the assumptions that the global cost function satisfies the restricted secant inequality condition and the gradients of each local cost function at optimal points are the same, which are also stronger than the P–Ł condition.

In many applications, such as optimal power flow problems, resource allocation problems, and empirical risk minimization problems, the cost functions are usually nonconvex. Thus, distributed nonconvex optimization has gained considerable attentions (see, e.g., [27]–[34]). In these studies, basic convergence results typically ensure that distributed algorithms converge to stationary points. For example, in [27], [29]–[32], and [34], it was shown that the first-order stationary point can be found with an $\mathcal{O}(1/T)$ convergence rate when each local cost function is smooth, where T is the total number of iterations.

Note that aforementioned distributed optimization algorithms use at least gradient information of the cost functions, and sometimes even second- or higher order information. However, in some practical applications, explicit expressions of the gradients are often unavailable or difficult to obtain [35]. For example, the cost functions of many big data problems that deal with complex data generating processes

Manuscript received 25 April 2021; accepted 21 August 2021. Date of publication 10 September 2021; date of current version 29 July 2022. This work was supported in part by the Knut and Alice Wallenberg Foundation, in part by the Swedish Foundation for Strategic Research, in part by the Swedish Research Council, in part by the National Natural Science Foundation of China under Grant 61991403, Grant 61991404, and Grant 61991400, and in part by the 2020 Science and Technology Major Project of Liaoning Province under Grant 2020JH1/10100008. A preliminary version of this article was presented at the 59th IEEE Conference on Decision and Control, Jeju Island, Republic of Korea, Dec. 14–18, 2020. Recommended by Associate Editor G. Notarstefano. *(Corresponding author: Tao Yang.)*

Xinlei Yi and Karl H. Johansson are with the Division of Decision and Control Systems, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, 114 28 Stockholm, Sweden, and also with the Digital Futures, 114 28 Stockholm, Sweden (e-mail: xinleiy@kth.se; kallej@kth.se).

Shengjun Zhang is with the Department of Electrical Engineering, University of North Texas, Denton, TX 76203 USA (e-mail: shengjunzhang@my.unt.edu).

Tao Yang and Tianyou Chai are with the State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang 110819, China (e-mail: taoyang.work@gmail.com; tychai@mail.neu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TAC.2021.3108501.

Digital Object Identifier 10.1109/TAC.2021.3108501

cannot be explicitly defined [36]. Thus, zeroth-order (gradient-free) optimization algorithms are needed. A key step in zeroth-order optimization algorithms is to estimate the gradient of the cost function by sampling the function values. Various gradient estimation methods have been developed (see, e.g., [37] and [38]). Some recent works have combined these gradient estimation methods with distributed first-order algorithms. For instance, the authors of [39]–[44] and [45], [46] proposed distributed zeroth-order algorithms for distributed convex and nonconvex optimization, respectively.

The main contribution of this article is on solving distributed nonconvex optimization problems. We first consider a distributed first-order primal–dual algorithm, which is a special form of the EXTRA algorithm proposed in [21]. When each local cost function is smooth, we show that it finds the first-order stationary point with a rate $\mathcal{O}(1/T)$ and that the cost difference between the global optimum and the resulting stationary point is bounded. We also show that not only the same algorithm can find a global optimum, but also the convergence rate is linear under an additional assumption that the global cost function satisfies the P–Ł condition. This condition is weaker than the (restrict) strong convexity condition assumed in [5]–[17], [21]–[24], and [26] since it does not require convexity and the global minimizer is not necessarily unique. This condition is also different from the metric subregularity criterion assumed in [25]. In other words, we show that for a larger class of cost functions than strongly convex functions, the global optimum can be founded linearly by the considered distributed algorithm. It should be highlighted that the P–Ł constant is not used to design the algorithm parameters. Noting that, generally, the P–Ł condition is difficult to check, with the above property that the P–Ł condition does not need to be checked when implementing the considered algorithm, which is a significant innovation. Another innovation is that the proofs of both sublinear and linear convergence are based on the same appropriately designed Lyapunov function, which facilitates extending our results to other settings, such as event-triggered communication. We notice that, recently, Xin *et al.* [47] considered a distributed randomized incremental gradient algorithm and achieved the same convergence results under the same conditions as ours. The algorithm considered in [47] is computationally efficient since it evaluates only one component gradient per agent per iteration. However, the P–Ł constant is used to design the stepsize in [47]. In other words, in order to implement the algorithm considered in [47], the P–Ł condition needs to be checked in advance, which is normally difficult in practice.

Motivated by the situation where the gradient information is unavailable, we then propose a distributed zeroth-order algorithm, by integrating the considered distributed first-order algorithm with the deterministic gradient estimator proposed in [38]. We show that it has the same convergence properties as the considered first-order algorithm under the same conditions. It should be mentioned that the analysis of both sublinear and linear convergence for our zeroth-order algorithm is based on the Lyapunov function modified from the Lyapunov function for the first-order algorithm. Compared with [46], which also proposed a distributed deterministic zeroth-order algorithm and established the same convergence properties under the same conditions as ours, one innovation of our zeroth-order algorithm is that the P–Ł constant, which is normally difficult to determine, is not used for designing the algorithm. Moreover, the proposed zeroth-order algorithm only requires each agent to communicate one p-dimensional variable with its neighbors at each iteration, where p is the dimension of the decision variable, while the algorithm proposed in [46] requires each agent to communicate three p-dimensional variables. The detailed comparison of this article to other related studies in the literature is summarized in tables provided in the online version [48] due to the space limitation.

The rest of this article is organized as follows. Section II introduces some preliminaries. Section III presents the problem formulation and assumptions. Sections IV and V provide the distributed first- and zeroth-order primal–dual algorithms and analyze their convergence properties, respectively. Simulations are given in Section VI. Finally, Section VII concludes this article. All the proofs are given in the online version [48] due to the space limitation.

Notations: $\mathbb{N}_0$ and $\mathbb{N}_+$ denote the set of nonnegative and positive integers, respectively. $\{\mathbf{e}_1,\ldots,\mathbf{e}_p\}$ represents the standard basis of $\mathbb{R}^p$. $[n]$ denotes the set $\{1,\ldots,n\}$ for any positive constant integer n. $\mathrm{col}(z_1,\ldots,z_k)$ is the concatenated column vector of vectors $z_i\in\mathbb{R}^{p_i}$, $i\in[k]$. $\mathbf{1}_n$ ($\mathbf{0}_n$) denotes the column one (zero) vector of dimension n. $\mathbf{I}_n$ is the n-dimensional identity matrix. Given a vector $[x_1,\ldots,x_n]^\top\in\mathbb{R}^n$, $\mathrm{diag}([x_1,\ldots,x_n])$ is a diagonal matrix with the ith diagonal element being x_i. The notation $A\otimes B$ denotes the Kronecker product of matrices A and B. $\mathrm{null}(A)$ is the null space of matrix A. $\rho(\cdot)$ stands for the spectral radius for matrices and $\rho_2(\cdot)$ indicates the minimum positive eigenvalue for matrices having positive eigenvalues. $\|\cdot\|$ represents the Euclidean norm for vectors or the induced 2-norm for matrices. For any square matrix A, denote $\|x\|_A^2=x^\top Ax$. Given a differentiable function f, ∇f denotes the gradient of f.

II. Preliminaries

In this section, we present some definitions and properties related to algebraic graph theory, the P–Ł condition, and the deterministic gradient estimator.

A. Algebraic Graph Theory

Let $\mathcal{G}=(\mathcal{V},\mathcal{E},A)$ denote a weighted undirected graph with the set of vertices (nodes) $\mathcal{V}=[n]$, the set of links (edges) $\mathcal{E}\subseteq\mathcal{V}\times\mathcal{V}$, and the weighted adjacency matrix $A=A^\top=(a_{ij})$ with nonnegative elements a_{ij}. A link of $\mathcal{G}$ is denoted by $(i,j)\in\mathcal{E}$ if $a_{ij}>0$, i.e., if vertices i and j can communicate with each other. It is assumed that $a_{ii}=0$ for all $i\in[n]$. Let $\mathcal{N}_i=\{j\in[n]:a_{ij}>0\}$ and $\deg_i=\sum_{j=1}^n a_{ij}$ denote the neighbor set and weighted degree of vertex i, respectively. The degree matrix of graph $\mathcal{G}$ is $\mathrm{Deg}=\mathrm{diag}([\deg_1,\ldots,\deg_n])$. The Laplacian matrix is $L=(L_{ij})=\mathrm{Deg}-A$. A path of length k between vertices i and j is a subgraph with distinct vertices $i_0=i,\ldots,i_k=j\in[n]$ and edges $(i_j,i_{j+1})\in\mathcal{E}$, $j=0,\ldots,k-1$. An undirected graph is connected if there exists at least one path between any two distinct vertices. If the graph $\mathcal{G}$ is connected, then its Laplacian matrix L is positive semidefinite and $\mathrm{null}(L)=\{\mathbf{1}_n\}$ (see [49]).

B. P–Ł Condition

Let $f(x):\mathbb{R}^p\mapsto\mathbb{R}$ be a differentiable function. Let $\mathbb{X}^*=\arg\min_{x\in\mathbb{R}^p}f(x)$ and $f^*=\min_{x\in\mathbb{R}^p}f(x)$. Moreover, we assume that $f^*>-\infty$.

Definition 1: The function f satisfies the P–Ł condition with constant $\nu>0$ if

$$\frac{1}{2}\|\nabla f(x)\|^2\geq\nu(f(x)-f^*)\quad\forall x\in\mathbb{R}^p. \tag{1}$$

It is straightforward to see that every (essentially or weakly) strongly convex function satisfies the P–Ł condition. The P–Ł condition implies that every stationary point is a global minimizer, i.e., $\mathbb{X}^*=\{x\in\mathbb{R}^p:\nabla f(x)=\mathbf{0}_p\}$. But unlike the (essentially or weakly) strong convexity, the P–Ł condition alone does not imply convexity of f. Moreover, it does not imply that $\mathbb{X}^*$ is a singleton either. The function $f(x)=x^2+3\sin^2(x)$ is an example of nonconvex functions satisfying the P–Ł condition with $\nu=1/32$ (see [20]).

C. Deterministic Gradient Estimator

Let $f(x):\mathbb{R}^p\mapsto\mathbb{R}$ be a differentiable function. Agarwal *et al.* [38] proposed the following deterministic gradient estimator:

$$\hat{\nabla}f(x,\delta)=\frac{1}{\delta}\sum_{l=1}^{p}(f(x+\delta\mathbf{e}_l)-f(x))\mathbf{e}_l \tag{2}$$

where $\delta>0$ is an exploration parameter. This gradient estimator can be calculated by sampling the function values of f at $p+1$ points. From [38, eq. (16)], we know that $\hat{\nabla}f(x,\delta)$ is close to $\nabla f(x)$ when δ is small, which is summarized in the following lemma.

Lemma 1: Suppose that f is smooth with constant L_f; then

$$\|\hat{\nabla}f(x,\delta)-\nabla f(x)\|\leq\frac{\sqrt{p}L_f\delta}{2}\quad\forall x\in\mathbb{R}^p,\ \forall\delta>0. \tag{3}$$

III. Problem Formulation and Assumptions

Consider a network of n agents, each of which has a private local cost function $f_i:\mathbb{R}^p\mapsto\mathbb{R}$. All agents collaborate to solve the following optimization problem:

$$\min_{x\in\mathbb{R}^p}f(x):=\frac{1}{n}\sum_{i=1}^{n}f_i(x). \tag{4}$$

The communication among agents is described by a weighted undirected graph $\mathcal{G}$. Let $\mathbb{X}^*$ and f^* denote the optimal set and the minimum function value of the optimization problem (4), respectively. The following assumptions are made.

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

Assumption 2: The optimal set $\mathbb{X}^*$ is nonempty and $f^*>-\infty$.

Assumption 3: Each local cost function $f_i(x)$ is smooth with constant $L_f>0$, i.e., $\|\nabla f_i(x)-\nabla f_i(y)\|\leq L_f\|x-y\|$, $\forall x,y\in\mathbb{R}^p$.

Assumption 4: The global cost function $f(x)$ satisfies the P–Ł condition with constant $\nu>0$.

Remark 1: Assumptions 1–3 are common in the literature, e.g., [5], [21]. Assumption 4 is weaker than the assumption that the global or each local cost function is strongly convex. It should be highlighted that the convexity of the cost functions and the boundedness of their gradients are not assumed. Moreover, we do not assume that $\mathbb{X}^*$ is a singleton or finite set either.

IV. Distributed First-Order Primal–Dual Algorithm

In this section, we consider a distributed first-order primal–dual algorithm and analyze its convergence property.

A. Algorithm Description

In this section, we present the derivation of the considered algorithm.

Denote $\boldsymbol{x}=\mathrm{col}(x_1,\ldots,x_n)$, $\tilde{f}(\boldsymbol{x})=\sum_{i=1}^n f_i(x_i)$, and $\boldsymbol{L}=L\otimes\mathbf{I}_p$. Recall that the Laplacian matrix L is positive semidefinite and $\mathrm{null}(L)=\{\mathbf{1}_n\}$ when $\mathcal{G}$ is connected. The optimization problem (4) is equivalent to the following constrained optimization problem:

$$\begin{aligned}\min_{\boldsymbol{x}\in\mathbb{R}^{np}}\quad&\tilde{f}(\boldsymbol{x})\\ \text{s.t.}\quad&\boldsymbol{L}^{1/2}\boldsymbol{x}=\mathbf{0}_{np}.\end{aligned} \tag{5}$$

Here, $\boldsymbol{L}^{1/2}=L^{1/2}\otimes\mathbf{I}_p$ and $L^{1/2}$ is the square root of the positive-semidefinite matrix L. Moreover, we use $\boldsymbol{L}^{1/2}\boldsymbol{x}=\mathbf{0}_{np}$ rather than $\boldsymbol{L}\boldsymbol{x}=\mathbf{0}_{np}$ as the constraint since they are both equivalent to $\boldsymbol{x}=\mathbf{1}_n\otimes x$ due to the fact that $\mathrm{null}(L^{1/2})=\mathrm{null}(L)=\{\mathbf{1}_n\}$, but the first has a particular property, which will be discussed in Remark 6.

Algorithm 1: Distributed First-Order Primal–Dual Algorithm.

Input: parameters $\alpha>0$, $\beta>0$, and $\eta>0$.
Initialize: $x_{i,0}\in\mathbb{R}^p$ and $v_{i,0}=\mathbf{0}_p$, $\forall i\in[n]$.
for $k=0,1,\ldots$ **do**
  for $i=1,\ldots,n$ in parallel **do**
    Broadcast $x_{i,k}$ to $\mathcal{N}_i$ and receive $x_{j,k}$ from $j\in\mathcal{N}_i$;
    Update $x_{i,k+1}$ by (9a);
    Update $v_{i,k+1}$ by (9b).
  end for
end for
Output: $\{\boldsymbol{x}_k\}$.

Let $\boldsymbol{u}\in\mathbb{R}^{np}$ denote the dual variable. Then, the augmented Lagrangian function associated with (5) is

$$\mathcal{A}(\boldsymbol{x},\boldsymbol{u})=\tilde{f}(\boldsymbol{x})+\frac{\alpha}{2}\boldsymbol{x}^\top\boldsymbol{L}\boldsymbol{x}+\beta\boldsymbol{u}^\top\boldsymbol{L}^{1/2}\boldsymbol{x} \tag{6}$$

where $\alpha>0$ and $\beta>0$ are the regularization parameters.

Based on the primal–dual gradient method, a distributed first-order algorithm to solve (5) is

$$\boldsymbol{x}_{k+1}=\boldsymbol{x}_k-\eta(\alpha\boldsymbol{L}\boldsymbol{x}_k+\beta\boldsymbol{L}^{1/2}\boldsymbol{u}_k+\nabla\tilde{f}(\boldsymbol{x}_k)) \tag{7a}$$

$$\boldsymbol{u}_{k+1}=\boldsymbol{u}_k+\eta\beta\boldsymbol{L}^{1/2}\boldsymbol{x}_k\quad\forall\boldsymbol{x}_0,\ \boldsymbol{u}_0\in\mathbb{R}^{np} \tag{7b}$$

where $\eta>0$ is a fixed stepsize. Denote $\boldsymbol{v}_k=\mathrm{col}(v_{1,k},\ldots,v_{n,k})=\boldsymbol{L}^{1/2}\boldsymbol{u}_k$. Then, the recursion (7) can be rewritten as

$$\boldsymbol{x}_{k+1}=\boldsymbol{x}_k-\eta(\alpha\boldsymbol{L}\boldsymbol{x}_k+\beta\boldsymbol{v}_k+\nabla\tilde{f}(\boldsymbol{x}_k)) \tag{8a}$$

$$\boldsymbol{v}_{k+1}=\boldsymbol{v}_k+\eta\beta\boldsymbol{L}\boldsymbol{x}_k\quad\forall\boldsymbol{x}_0\in\mathbb{R}^{np},\ \sum_{j=1}^{n}v_{j,0}=\mathbf{0}_p. \tag{8b}$$

The initialization condition $\sum_{j=1}^n v_{j,0}=\mathbf{0}_p$ is derived from $\boldsymbol{v}_0=\boldsymbol{L}^{1/2}\boldsymbol{u}_0$, and it is easy to be satisfied, for example, $v_{i,0}=\mathbf{0}_p$, $\forall i\in[n]$, or $v_{i,0}=\sum_{j=1}^n L_{ij}x_{j,0}$, $\forall i\in[n]$. It is straightforward to verify that the algorithm (8) is a special form of the EXTRA algorithm proposed in [21] with mixing matrices $\boldsymbol{W}=\mathbf{I}_{np}-\eta\alpha\boldsymbol{L}$ and $\tilde{\boldsymbol{W}}=\mathbf{I}_{np}-\eta\alpha\boldsymbol{L}+\eta^2\beta^2\boldsymbol{L}$. It is also a special form of general frameworks for primal–dual decentralized algorithms studied in [50]. Note that (8) can be written agentwise as

$$x_{i,k+1}=x_{i,k}-\eta\left(\alpha\sum_{j=1}^{n}L_{ij}x_{j,k}+\beta v_{i,k}+\nabla f_i(x_{i,k})\right) \tag{9a}$$

$$v_{i,k+1}=v_{i,k}+\eta\beta\sum_{j=1}^{n}L_{ij}x_{j,k},\ \forall x_{i,0}\in\mathbb{R}^p,\ \sum_{j=1}^{n}v_{j,0}=\mathbf{0}_p. \tag{9b}$$

This corresponds to the considered distributed first-order primal–dual algorithm, which is presented in pseudo-code as Algorithm 1.

Remark 2: In the literature, various distributed first-order algorithms have been proposed to solve the nonconvex optimization problem (4), for example, distributed gradient descent (DGD) algorithm [28], [34], distributed gradient tracking algorithm [34], distributed algorithm based on a novel approximate filtering-then-predict and tracking (xFILTER) strategy [31]. Compared with the considered distributed algorithm (9), these algorithms have some potential drawbacks. For the DGD algorithm, existing studies, such as [28] and [34], only showed that the output of the algorithm converges to a neighborhood of a stationary point unless additional assumptions, such as the boundedness of the gradients of cost functions, are assumed. The xFILTER algorithm [31] is a double-loop algorithm and, thus, more complicated than (9).

B. Convergence Analysis

In this section, we provide convergence analysis for both scenarios without and with Assumption 4.

Denote $K_n = \mathbf{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top$, $\boldsymbol{K} = K_n \otimes \mathbf{I}_p$, $\boldsymbol{H} = \frac{1}{n}(\mathbf{1}_n\mathbf{1}_n^\top \otimes \mathbf{I}_p)$, $\bar{x}_k = \frac{1}{n}(\mathbf{1}_n^\top \otimes \mathbf{I}_p)\boldsymbol{x}_k$, $\bar{\boldsymbol{x}}_k = \mathbf{1}_n \otimes \bar{x}_k$, $\boldsymbol{g}_k = \nabla\tilde{f}(\boldsymbol{x}_k)$, $\bar{\boldsymbol{g}}_k = \boldsymbol{H}\boldsymbol{g}_k$, $\boldsymbol{g}_k^0 = \nabla\tilde{f}(\bar{\boldsymbol{x}}_k)$, $\bar{\boldsymbol{g}}_k^0 = \boldsymbol{H}\boldsymbol{g}_k^0 = \mathbf{1}_n \otimes \nabla f(\bar{x}_k)$, and

$$\hat{V}_k = \|\boldsymbol{x}_k\|_{\boldsymbol{K}}^2 + \|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\|_{\boldsymbol{K}}^2 + n(f(\bar{x}_k) - f^*)$$

$$W_k = \|\boldsymbol{x}_k - \bar{\boldsymbol{x}}_k\|^2 + \|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\|_{\boldsymbol{K}}^2 + \|\bar{\boldsymbol{g}}_k\|^2 + \|\bar{\boldsymbol{g}}_k^0\|^2.$$

We have the following convergence result for Algorithm 1 without Assumption 4.

Theorem 1: Suppose that Assumptions 1–3 hold. Let $\{\boldsymbol{x}_k\}$ be the sequence generated by Algorithm 1 with $\alpha \in (\beta + \kappa_1, \kappa_2\beta]$, $\beta > \max\{\frac{\kappa_1}{\kappa_2 - 1}, \kappa_3, \kappa_4\}$, and $\eta \in (0, \min\{\frac{\epsilon_1}{\epsilon_2}, \frac{\epsilon_3}{\epsilon_4}, \frac{\epsilon_5}{\epsilon_6}\})$. Then, we have

$$\sum_{k=0}^{T} W_k \leq \frac{\epsilon_8 \hat{V}_0}{\epsilon_7} \quad \forall T \in \mathbb{N}_0 \tag{10}$$

$$f(\bar{x}_{T+1}) - f^* \leq \frac{\epsilon_8 \hat{V}_0}{n} \quad \forall T \in \mathbb{N}_0 \tag{11}$$

where

$$\kappa_1 = \frac{1}{2\rho_2(L)}(2 + 3L_f^2),\ \kappa_2 > 1$$

$$\kappa_3 = \frac{1}{4}\left(1 + \left(1 + 8\kappa_2 + \frac{8}{\rho_2(L)}\right)^{\frac{1}{2}}\right)$$

$$\kappa_4 = \left(\kappa_2 + \frac{1}{\rho_2(L)}\right)L_f^2 + \left(\left(\kappa_2 + \frac{1}{\rho_2(L)}\right)^2 L_f^2 + 2\right)^{\frac{1}{2}} L_f$$

$$\epsilon_1 = (\alpha - \beta)\rho_2(L) - \frac{1}{2}(2 + 3L_f^2)$$

$$\epsilon_2 = \beta^2\rho(L) + (2\alpha^2 + \beta^2)\rho^2(L) + \frac{5}{2}L_f^2$$

$$\epsilon_3 = \beta - \frac{1}{2} - \frac{\alpha}{2\beta^2} - \frac{1}{2\beta\rho_2(L)},\ \epsilon_4 = 2\beta^2 + \frac{1}{2}$$

$$\epsilon_5 = \frac{1}{4} - \frac{1}{2\beta}\left(\frac{1}{\beta} + \frac{1}{\rho_2(L)} + \frac{\alpha}{\beta}\right)L_f^2$$

$$\epsilon_6 = \frac{1}{\beta^2}\left(1 + \frac{1}{\rho_2(L)} + \frac{\alpha}{\beta}\right)L_f^2 + \frac{L_f(1 + L_f)}{2}$$

$$\epsilon_7 = \eta \min\left\{\epsilon_1 - \eta\epsilon_2,\ \epsilon_3 - \eta\epsilon_4,\ \epsilon_5 - \eta\epsilon_6,\ \frac{1}{4}\right\}$$

$$\epsilon_8 = \frac{\alpha + \beta}{2\beta} + \frac{1}{2\rho_2(L)}.$$

Remark 3: We should point out that the settings on the parameters α, β, and η are just sufficient conditions. With some modifications of the proofs, other forms of settings for these algorithm parameters still can guarantee the same kind of convergence rate. Moreover, the interval $(\beta + \kappa_1, \kappa_2\beta]$ is nonempty due to the settings that $\beta > \kappa_1/(\kappa_2 - 1)$ and $\kappa_2 > 1$. From (10), we know that $\min_{k\in[T]}\{\|\boldsymbol{x}_k - \bar{\boldsymbol{x}}_k\|^2 + \|\bar{\boldsymbol{g}}_k^0\|^2\} = \mathcal{O}(1/T)$. In other words, Algorithm 1 finds a stationary point of the nonconvex optimization problem (4) with a rate $\mathcal{O}(1/T)$. This rate is the same as that achieved by the distributed gradient tracking algorithm [34] and the xFILTER algorithm [31] under the same assumptions on the cost functions. From (11), we know that the cost difference between the global optimum and the resulting stationary point is bounded.

With Assumption 4, the following result states that Algorithm 1 can find a global optimum and the convergence rate is linear.

Theorem 2: Suppose that Assumptions 1–4 hold. Let $\{\boldsymbol{x}_k\}$ be the sequence generated by Algorithm 1 with the same α, β, and η given in Theorem 1. Then, we have

$$\|\boldsymbol{x}_k - \bar{\boldsymbol{x}}_k\|^2 + n(f(\bar{x}_k) - f^*) \leq (1 - \epsilon)^k c \quad \forall k \in \mathbb{N}_0 \tag{12}$$

where

$$\epsilon = \frac{\epsilon_{10}}{\epsilon_8} \in (0, 1),\ c = \frac{\epsilon_8 \hat{V}_0}{\epsilon_9},\ \epsilon_9 = \min\left\{\frac{1}{2\rho(L)}, \frac{\alpha - \beta}{2\alpha}\right\}$$

$$\epsilon_{10} = \eta \min\left\{\epsilon_1 - \eta\epsilon_2, \epsilon_3 - \eta\epsilon_4, \frac{\nu}{2}\right\}.$$

Moreover, if the projection operator $\mathcal{P}_{\mathbb{X}^*}(\cdot)$ is well defined, then

$$\|\boldsymbol{x}_k - \mathbf{1}_n \otimes \mathcal{P}_{\mathbb{X}^*}(\bar{x}_k)\|^2 \leq (1 - \epsilon)^k c\left(1 + \frac{1}{2\nu}\right) \quad \forall k \in \mathbb{N}_0. \tag{13}$$

Remark 4: From Theorems 1 and 2, we know that the considered first-order primal–dual algorithm uses the same algorithm parameters for the cases without and with the P–Ł condition. The proofs of both theorems are based on the same appropriately designed Lyapunov function given in the proof. In the literature that considered distributed nonconvex optimization, e.g., [29]–[33], the lower bounded potential functions (which may be negative) are commonly used to analyze the convergence properties of the proposed algorithms. Therefore, the analysis in those studies cannot be extended to show linear convergence when the P–Ł condition holds since the lower bounded potential functions may not be Lyapunov functions. In the literature that obtained linear convergence for distributed optimization, e.g., [5]–[17], [21]–[26], the convexity and/or the uniqueness of the global minimizer are the key in the analysis. Therefore, the analysis in those studies cannot be extended to show linear convergence when strong convexity is relaxed by the P–Ł condition since the later does not imply convexity of cost functions and the uniqueness of the global minimizers.

Remark 5: The distributed first-order algorithms proposed in [5]–[17] and [21]–[26] also established linear convergence. However, in [5]–[14], it was assumed that each local cost function is strongly convex. In [15] and [16], it was assumed that each local cost function is convex and the global cost function is strongly convex. In [17], it was assumed that the global cost function is strongly convex. In [21] and [24], it was assumed that each local cost function is convex, the global cost function is restricted strongly convex, and $\mathbb{X}^*$ is a singleton. In [22] and [23], it was assumed that each local cost function is restricted strongly convex and the optimal set $\mathbb{X}^*$ is a singleton. In [25], it was assumed that each local cost function is convex and the primal–dual gradient map is metrically subregular. In [26], it was assumed that the global cost function satisfies the restricted secant inequality condition and the gradients of each local cost function at optimal points are the same. In contrast, the linear convergence result established in Theorem 2 only requires that the global cost function satisfies the P–Ł condition, but the convexity assumption on cost functions and the singleton assumption on the optimal set and the set of each local cost function's gradients at the optimal points are not required. Moreover, it should be highlighted that the P–Ł constant ν is not used when implementing Algorithm 1. This is an important property since it is normally difficult to determine the P–Ł constant. Compared with some of the aforementioned studies, one potential drawback is that we assume that the communication graph is static and undirected. We leave the extension to time-varying directed graph for future work.

Algorithm 2 Distributed Deterministic Zeroth-Order Primal–Dual Algorithm

Input: parameters $\alpha > 0$, $\beta > 0$, $\eta > 0$, and $\{\delta_{i,k} > 0\}$.
Initialize: $x_{i,0} \in \mathbb{R}^p$ and $v_{i,0} = \mathbf{0}_p$, $\forall i \in [n]$.
for $k = 0, 1, \ldots$ **do**
  for $i = 1, \ldots, n$ in parallel **do**
    Broadcast $x_{i,k}$ to $\mathcal{N}_i$ and receive $x_{j,k}$ from $j \in \mathcal{N}_i$;
    Sample $f_i(x_{i,k})$ and $\{f_i(x_{i,k} + \delta_{i,k}\mathbf{e}_l)\}_{l=1}^p$;
    Update $x_{i,k+1}$ by (15a);
    Update $v_{i,k+1}$ by (15b).
  end for
end for
Output: $\{\boldsymbol{x}_k\}$.

Remark 6: If we use $\boldsymbol{L}\boldsymbol{x} = \mathbf{0}_{np}$ as the constraint in (5), then we can construct an alternative distributed primal–dual gradient descent algorithm

$$x_{i,k+1} = x_{i,k} - \eta\left(\sum_{j=1}^n L_{ij}(\alpha x_{j,k} + \beta v_{j,k}) + \nabla f_i(x_{i,k})\right) \quad (14a)$$

$$v_{i,k+1} = v_{i,k} + \eta\beta\sum_{j=1}^n L_{ij}x_{j,k} \quad \forall x_{i,0},\, v_{i,0} \in \mathbb{R}^p. \quad (14b)$$

Similar results as shown in Theorems 1 and 2 (as well as the results stated in Theorems 3 and 4 in the next section) can be obtained. We omit the details due to the space limitation. Different from the requirement that $\sum_{j=1}^n v_{j,0} = \mathbf{0}_p$ in the algorithm (9), $v_{i,0}$ can be arbitrarily chosen in the algorithm (14). In other words, the algorithm (14) is robust to the initial condition $v_{i,0}$. However, it requires additional communication of $v_{j,k}$ in (14a), compared to (9).

V. Distributed Deterministic Zeroth-Order Primal–Dual Algorithm

In this section, we propose a distributed deterministic zeroth-order primal–dual algorithm and analyze its convergence property.

A. Algorithm Description

When implementing the first-order algorithm (9), each agent needs to know the gradient of its local cost function. However, in some practical applications, the explicit expressions of the gradients are unavailable or difficult to obtain [35]. Inspired by the deterministic gradient estimator (2), based on the considered distributed first-order algorithm (9), we propose the following zeroth-order algorithm:

$$x_{i,k+1} = x_{i,k} - \eta\left(\alpha\sum_{j=1}^n L_{ij}x_{j,k} + \beta v_{i,k} + \hat{\nabla} f_i(x_{i,k}, \delta_{i,k})\right) \quad (15a)$$

$$v_{i,k+1} = v_{i,k} + \eta\beta\sum_{j=1}^n L_{ij}x_{j,k} \quad \forall x_{i,0} \in \mathbb{R}^p,\, \sum_{j=1}^n v_{j,0} = \mathbf{0}_p \quad (15b)$$

where $\hat{\nabla} f_i(x_{i,k}, \delta_{i,k})$ is the deterministic estimator of $\nabla f_i(x_{i,k})$, as defined in (2). Note that the gradient estimator $\hat{\nabla} f_i(x_{i,k}, \delta_{i,k})$ can be calculated by sampling the function values of f_i at $p+1$ points.

We present the distributed deterministic zeroth-order primal–dual algorithm (15) in pseudocode as Algorithm 2.

Remark 7: A different distributed deterministic zeroth-order algorithm was proposed in [46]. However, in that algorithm, at each iteration, each agent i needs to communicate two additional p-dimensional variables besides the communication of $x_{i,k}$ with its neighbors, which results in a heavy communication burden when p is large. Moreover, the deterministic gradient estimator used in [46] requires that at each iteration, each agent samples its local cost function values at $2p$ points compared with $p+1$ points used in our algorithm.

B. Convergence Analysis

In this section, we provide convergence analysis for both scenarios without and with Assumption 4.

We use the same notations introduced in Section IV. Moreover, denote $h_{i,k} = \hat{\nabla} f_i(x_{i,k}, \delta_{i,k})$, $\boldsymbol{h}_k = \mathrm{col}(h_{1,k}, \ldots, h_{n,k})$, $\bar{\boldsymbol{h}}_k = \boldsymbol{H}\boldsymbol{h}_k$, $\delta_k = \max_{i\in[n]}\{\delta_{i,k}\}$, $h^0_{i,k} = \hat{\nabla} f_i(\bar{x}_k, \delta_k)$, $\boldsymbol{h}^0_k = \mathrm{col}(h^0_{1,k}, \ldots, h^0_{n,k})$, $\bar{\boldsymbol{h}}^0_k = \boldsymbol{H}\boldsymbol{h}^0_k$, and

$$\hat{U}_k = \|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} + \|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{h}^0_k\|^2_{\boldsymbol{K}} + n(f(\bar{x}_k) - f^*).$$

We have the following convergence result for Algorithm 2 without Assumption 4.

Theorem 3: Suppose that Assumptions 1–3 hold. Let $\{\boldsymbol{x}_k\}$ be the sequence generated by Algorithm 2 with $\alpha \in (\beta + \tilde{\kappa}_1, \kappa_2\beta]$, $\beta > \max\{\frac{\tilde{\kappa}_1}{\kappa_2 - 1}, \kappa_3, \tilde{\kappa}_4\}$, $\eta \in (0, \min\{\frac{\tilde{\epsilon}_1}{\tilde{\epsilon}_2}, \frac{\epsilon_3}{\epsilon_4}, \frac{\tilde{\epsilon}_5}{\tilde{\epsilon}_6}\})$, and $\delta_{i,k} > 0$ such that

$$\delta^a_i = \sum_{k=0}^{+\infty}\delta^2_{i,k} < +\infty. \quad (16)$$

Then

$$\sum_{k=0}^T(\|\boldsymbol{x}_k - \bar{\boldsymbol{x}}_k\|^2 + \|\bar{\boldsymbol{g}}^0_k\|^2) \le \frac{\tilde{c}}{\tilde{\epsilon}_7} \quad \forall T \in \mathbb{N}_0 \quad (17)$$

$$f(\bar{x}_{T+1}) - f^* \le \frac{\tilde{c}}{n} \quad \forall T \in \mathbb{N}_0 \quad (18)$$

where

$$\tilde{\kappa}_1 = \frac{1}{2\rho_2(L)}(2 + 9L_f^2)$$

$$\tilde{\kappa}_4 = 6\left(\kappa_2 + \frac{1}{\rho_2(L)}\right)L_f^2 + 2\left(9\left(\kappa_2 + \frac{1}{\rho_2(L)}\right)^2 L_f^4 + 3L_f^2\right)^{\frac{1}{2}}$$

$$\tilde{\epsilon}_1 = (\alpha - \beta)\rho_2(L) - \frac{1}{2}(2 + 9L_f^2)$$

$$\tilde{\epsilon}_2 = \beta^2\rho(L) + (2\alpha^2 + \beta^2)\rho^2(L) + \frac{15}{2}L_f^2$$

$$\tilde{\epsilon}_5 = \frac{1}{8} - \frac{3}{2\beta}\left(\frac{1}{\beta} + \frac{1}{\rho_2(L)} + \frac{\alpha}{\beta}\right)L_f^2$$

$$\tilde{\epsilon}_6 = \frac{3}{\beta^2}\left(1 + \frac{1}{\rho_2(L)} + \frac{\alpha}{\beta}\right)L_f^2 + \frac{L_f(1 + 3L_f)}{2}$$

$$\tilde{\epsilon}_7 = \eta\min\left\{\tilde{\epsilon}_1 - \eta\tilde{\epsilon}_2, \frac{1}{8}\right\}, \epsilon_{11} = \left(\frac{15\eta}{4} + 5\eta^2\right)\frac{3npL_f^2}{4} + \epsilon_{12}$$

$$\epsilon_{12} = \left(\left(\frac{1}{\beta^2} + \frac{1}{2\eta\beta}\right)\left(\frac{1}{\rho_2(L)} + \frac{\alpha}{\beta}\right) + \frac{1}{2\eta\beta^2} + \frac{1}{\beta^2} + \frac{1}{2}\right)\frac{3npL_f^2}{4}$$

$$\tilde{c} = \epsilon_8\hat{U}_0 + (\epsilon_{11} + \epsilon_{12})\sum_{i=1}^n\delta^a_i.$$

Remark 8: Similar to the discussion in Remark 3, from (17), we know that Algorithm 2 finds a stationary point of the nonconvex optimization problem (4) with a rate $\mathcal{O}(1/T)$. This rate is the same as that achieved by the distributed stochastic zeroth-order algorithm proposed in [45] under different assumptions. More specifically, Hajinezhad *et al.* [45] consider a more realistic scenario, where the cost function values are queried with noises. However, in [45], it needs an additional assumption that the gradient of each local cost function is bounded and each agent needs to employ $\mathcal{O}(T)$ function value samplings at each iteration. From (18), we know that the cost difference between the global optimum and the resulting stationary point is bounded.

With Assumption 4, the following result states that Algorithm 2 can find a global optimum and the convergence rate is linear.

Theorem 4: Suppose that Assumptions 1–4 hold. Let $\{\boldsymbol{x}_k\}$ be the sequence generated by Algorithm 2 with the same α, β, and η given in Theorem 3, and $\delta_{i,k} \in (0, \hat{\epsilon}^{\frac{k}{2}}]$; then

$$\|\boldsymbol{x}_k - \bar{\boldsymbol{x}}_k\|^2 + n(f(\bar{x}_k) - f^*) \leq \frac{1}{\epsilon_9}((1-\tilde{\epsilon})^{k+1}\epsilon_8\hat{U}_0 + \phi(\tilde{\epsilon}, \hat{\epsilon}, \check{\epsilon})) \quad \forall k \in \mathbb{N}_0 \tag{19}$$

where $\hat{\epsilon} \in (0,1)$, $\check{\epsilon} \in (\hat{\epsilon}, 1)$,

$$\tilde{\epsilon} = \frac{\tilde{\epsilon}_{10}}{\epsilon_8} \in (0,1),\ \tilde{\epsilon}_{10} = \eta \min\left\{\tilde{\epsilon}_1 - \eta\tilde{\epsilon}_2,\ \epsilon_3 - \eta\epsilon_4,\ \frac{\nu}{4}\right\}$$

$$\phi(\tilde{\epsilon}, \hat{\epsilon}, \check{\epsilon}) = \left(\frac{\epsilon_{11}}{1-\tilde{\epsilon}} + \epsilon_{12}\right)\begin{cases} \dfrac{(1-\tilde{\epsilon})^{k+1}}{1-\tilde{\epsilon}-\hat{\epsilon}}, & \text{if } 1-\tilde{\epsilon} > \hat{\epsilon} \\ \dfrac{\hat{\epsilon}^{k+1}}{\hat{\epsilon}+\tilde{\epsilon}-1}, & \text{if } 1-\tilde{\epsilon} < \hat{\epsilon} \\ \dfrac{\check{\epsilon}^{k+1}}{\check{\epsilon}-\hat{\epsilon}}, & \text{if } 1-\tilde{\epsilon} = \hat{\epsilon}. \end{cases}$$

Moreover, if the projection operator $\mathcal{P}_{\mathbb{X}^*}(\cdot)$ is well defined, then

$$\|\boldsymbol{x}_k - \mathbf{1}_n \otimes \mathcal{P}_{\mathbb{X}^*}(\bar{x}_k)\|^2 \leq \frac{1}{\epsilon_9}\left(1 + \frac{1}{2\nu}\right)((1-\epsilon)^{k+1}\epsilon_8\hat{U}_0 + \phi(\tilde{\epsilon}, \hat{\epsilon}, \check{\epsilon})) \quad \forall k \in \mathbb{N}_0. \tag{20}$$

Remark 9: It is straightforward to see that $\phi = \mathcal{O}(a^k)$, where $a = \max\{1-\tilde{\epsilon}, \hat{\epsilon}, \check{\epsilon}\} < 1$, so ϕ linearly converges to zero. By comparing Theorems 1 and 2 with Theorems 3 and 4, respectively, we see that the considered distributed first- and zeroth-order algorithms have the same convergence properties under the same assumptions. Similar convergence results as stated in Theorems 3 and 4 were also achieved by the distributed deterministic zeroth-order algorithm proposed in [46] under the same assumptions. Compared with [46], in addition to the advantages discussed in Remark 7, one more potential advantage of Theorem 4 is that the P–Ł constant ν is not used. However, Tang *et al.* [46] also proposed a distributed random zeroth-order algorithm. We expect that the considered distributed first-order algorithm (9) can be extended to be a random zeroth-order algorithm with two noisy samples of local cost function values by each agent at each iteration. Such an extension is our ongoing article.

VI. Simulations

In this section, we verify and illustrate the theoretical results through numerical simulations.

We consider the nonconvex distributed binary classification problem in [31], which is formulated as the optimization problem (4) with each component function f_i given by

$$f_i(x) = \frac{1}{m}\sum_{s=1}^{m}\log(1 + e^{-y_{is}x^\top z_{is}}) + \sum_{l=1}^{p}\frac{\lambda\mu([x]_l)^2}{1+\mu([x]_l)^2}$$

where m is the number of data points of each sensor, $y_{is} \in \{1, -1\}$ denotes the label for the sth data point of sensor i, $z_{is} \in \mathbb{R}^p$ is the feature vector, and λ and μ are regularization parameters. All settings for cost functions and the communication graph are the same as those described in [31]. Specifically, $n = 20$, $p = 50$, $m = 200$, $\lambda = 0.001$, and $\mu = 1$. The graph used in the simulation is the random geometric graph, and the graph parameter is set to be 0.5. We independently and randomly generate nm data points with dimension p and each agent contains m data points.

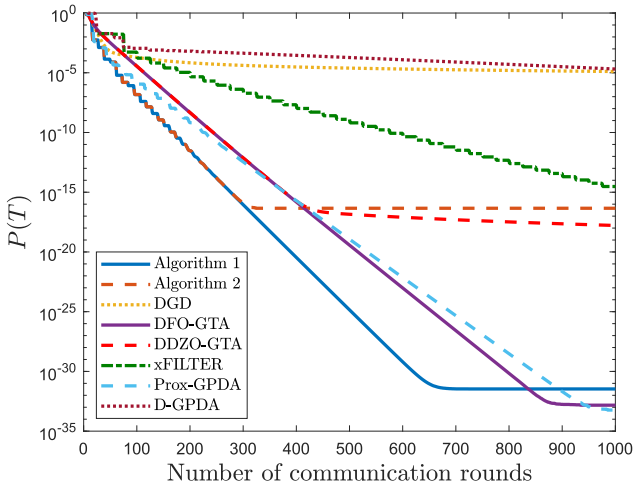

Fig. 1. Evolutions of $P(T)$ with respect to the number of communication rounds.

We compare Algorithms 1 and 2 with state-of-the-art algorithms: DGD with diminishing stepsizes [28], [34], distributed first-order gradient tracking algorithm (DFO-GTA) [11], [34], distributed deterministic zeroth-order gradient tracking algorithm (DDZO-GTA) [46], xFILTER [31], proximal gradient primal–dual algorithm [29], and distributed gradient primal–dual algorithm [30].

We use $P(T) = \min_{k\in[T]}\{\|\nabla f(\bar{x}_k)\|^2 + \frac{1}{n}\sum_{i=1}^{n}\|x_{i,k} - \bar{x}_k\|^2\}$ to measure the performance of each algorithm. Fig. 1 illustrates the convergence of $P(T)$ with respect to the number of communication rounds T for these algorithms with the same initial condition. It can be seen that the first-order algorithm (see Algorithm 1) gives the best performance in general. We also see that both zeroth-order algorithms (Algorithm 2 and DDZO-GTA [46]) exhibit almost identical behavior as their first-order counterparts (Algorithm 1 and DFO-GTA [11], [34]) during the early stage, but then slow down and converge at a sublinear rate.

In order to compare the performance of the two deterministic zeroth-order algorithms (Algorithm 2 and DDZO-GTA [46]), we plot the convergence of $P(T)$ with respect to the number of function value queries and variables communicated in Fig. 2. It can be seen that Algorithm 2 gives better performance.

VII. Conclusion

In this article, we studied distributed nonconvex optimization. We considered distributed first- and zeroth-order primal–dual algorithms and derived their convergence properties. Linear convergence was

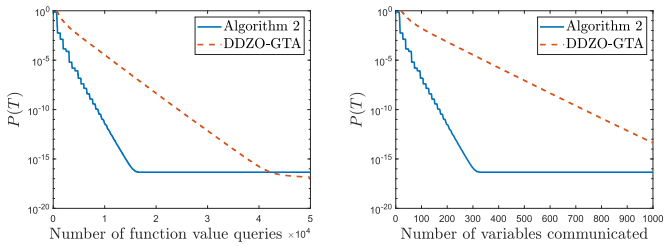

Fig. 2. Evolutions of $P(T)$ with respect to the number of function value queries (left) and the number of variables communicated (right).

established when the global cost function satisfies the P–Ł condition. This relaxes the standard strong convexity condition in the literature. Interesting directions for future work include proving linear convergence for larger stepsizes, considering time-varying graphs, investigating the scenarios where the function values are sampled with noises, and studying constraints.

Acknowledgment

The authors would like to thank Dr. Mingyi Hong, Dr. Na Li, Dr. Haoran Sun, and Dr. Yujie Tang for sharing their codes.

References

[1] J. N. Tsitsiklis, "Problems in decentralized decision making and computation," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, USA, 1984.
[2] T. Yang *et al.*, "A survey of distributed optimization," *Annu. Rev. Control*, vol. 47, pp. 278–305, 2019.
[3] B. Johansson, T. Keviczky, M. Johansson, and K. H. Johansson, "Subgradient methods and consensus algorithms for solving convex optimization problems," in *Proc. IEEE Conf. Decis. Control*, 2008, pp. 4185–4190.
[4] T. Yang *et al.*, "A distributed algorithm for economic dispatch over time-varying directed networks with delays," *IEEE Trans. Ind. Electron.*, vol. 64, no. 6, pp. 5095–5106, Jun. 2017.
[5] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the linear convergence of the ADMM in decentralized consensus optimization," *IEEE Trans. Signal Process.*, vol. 62, no. 7, pp. 1750–1761, Apr. 2014.
[6] D. Jakovetić, J. M. Moura, and J. Xavier, "Linear convergence rate of a class of distributed augmented lagrangian algorithms," *IEEE Trans. Autom. Control*, vol. 60, no. 4, pp. 922–936, Apr. 2015.
[7] A. Makhdoumi and A. Ozdaglar, "Convergence rate of distributed ADMM over networks," *IEEE Trans. Autom. Control*, vol. 62, no. 10, pp. 5082–5095, Oct. 2017.
[8] A. S. Berahas, R. Bollapragada, N. S. Keskar, and E. Wei, "Balancing communication and computation in distributed optimization," *IEEE Trans. Autom. Control*, vol. 64, no. 8, pp. 3141–3155, Aug. 2019.
[9] A. Nedić, A. Olshevsky, and W. Shi, "Achieving geometric convergence for distributed optimization over time-varying graphs," *SIAM J. Optim.*, vol. 27, no. 4, pp. 2597–2633, 2017.
[10] G. Qu and N. Li, "Accelerated distributed Nesterov gradient descent," *IEEE Trans. Autom. Control*, vol. 65, no. 6, pp. 2566–2581, Jun. 2020.
[11] G. Qu and N. Li, "Harnessing smoothness to accelerate distributed optimization," *IEEE Control Netw. Syst.*, vol. 5, no. 3, pp. 1245–1260, Sep. 2018.
[12] D. Jakovetić, "A unification and generalization of exact distributed first-order methods," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 5, no. 1, pp. 31–46, Mar. 2019.
[13] C. Xi, R. Xin, and U. A. Khan, "ADD-OPT: Accelerated distributed directed optimization," *IEEE Trans. Autom. Control*, vol. 63, no. 5, pp. 1329–1339, May 2018.
[14] J. Xu, S. Zhu, Y. C. Soh, and L. Xie, "Convergence of asynchronous distributed gradient methods over stochastic networks," *IEEE Trans. Autom. Control*, vol. 63, no. 2, pp. 434–448, Feb. 2018.
[15] D. Varagnolo, F. Zanella, A. Cenedese, G. Pillonetto, and L. Schenato, "Newton-Raphson consensus for distributed convex optimization," *IEEE Trans. Autom. Control*, vol. 61, no. 4, pp. 994–1009, Apr. 2016.
[16] M. Maros and J. Jaldén, "On the Q-linear convergence of distributed generalized ADMM under non-strongly convex function components," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 5, no. 3, pp. 442–453, Sep. 2019.
[17] Y. Tian, Y. Sun, and G. Scutari, "ASY-SONATA: Achieving linear convergence in distributed asynchronous multiagent optimization," in *Proc. Annu. Allerton Conf. Commun., Control, Comput.*, 2018, pp. 543–551.
[18] T. Yang, J. George, J. Qin, X. Yi, and J. Wu, "Distributed finite-time least squares solver for network linear equations," *Automatica*, vol. 113, 2020, Art. no. 108798.
[19] I. Necoara, Y. Nesterov, and F. Glineur, "Linear convergence of first order methods for non-strongly convex optimization," *Math. Program.*, vol. 175, nos. 1/2, pp. 69–107, 2019.
[20] H. Karimi, J. Nutini, and M. Schmidt, "Linear convergence of gradient and proximal-gradient methods under the Polyak-Łojasiewicz condition," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2016, pp. 795–811.
[21] W. Shi, Q. Ling, G. Wu, and W. Yin, "EXTRA: An exact first-order algorithm for decentralized consensus optimization," *SIAM J. Optim.*, vol. 25, no. 2, pp. 944–966, 2015.
[22] J. Zeng and W. Yin, "Extrapush for convex smooth decentralized optimization over directed networks," *J. Comput. Math.*, vol. 35, no. 4, pp. 383–396, 2017.
[23] C. Xi and U. A. Khan, "DEXTRA: A fast algorithm for optimization over directed graphs," *IEEE Trans. Autom. Control*, vol. 62, no. 10, pp. 4980–4993, Oct. 2017.
[24] X. Yi, L. Yao, T. Yang, J. George, and K. H. Johansson, "Distributed optimization for second-order multi-agent systems with dynamic event-triggered communication," in *Proc. IEEE Conf. Decis. Control*, 2018, pp. 3397–3402.
[25] S. Liang, L. Y. Wang, and G. Yin, "Exponential convergence of distributed primal-dual convex optimization algorithm without strong convexity," *Automatica*, vol. 105, pp. 298–306, 2019.
[26] X. Yi, S. Zhang, T. Yang, T. Chai, and K. H. Johansson, "Exponential convergence for distributed smooth optimization under the restricted secant inequality condition," in *Proc. IFAC World Congr.*, 2020, pp. 2672–2677.
[27] T. Tatarenko and B. Touri, "Non-convex distributed optimization," *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 3744–3757, Aug. 2017.
[28] J. Zeng and W. Yin, "On nonconvex decentralized gradient descent," *IEEE Trans. Signal Process.*, vol. 66, no. 11, pp. 2834–2848, Jun. 2018.
[29] M. Hong, D. Hajinezhad, and M.-M. Zhao, "Prox-PDA: The proximal primal-dual algorithm for fast distributed nonconvex optimization and learning over networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1529–1538.
[30] H. Sun and M. Hong, "Distributed non-convex first-order optimization and information processing: Lower complexity bounds and rate optimal algorithms," in *Proc. 52nd Asilomar Conf. Signals, Syst., Comput.*, 2018, pp. 38–42.
[31] H. Sun and M. Hong, "Distributed non-convex first-order optimization and information processing: Lower complexity bounds and rate optimal algorithms," *IEEE Trans. Signal Process.*, vol. 67, no. 22, pp. 5912–5928, Nov. 2019.
[32] D. Hajinezhad and M. Hong, "Perturbed proximal primal-dual algorithm for nonconvex nonsmooth optimization," *Math. Program.*, vol. 176, no. 1/2, pp. 207–245, 2019.
[33] M. Hong, M. Razaviyayn, and J. Lee, "Gradient primal-dual algorithm converges to second-order stationary solution for nonconvex distributed optimization over networks," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 2009–2018.
[34] A. Daneshmand, G. Scutari, and V. Kungurtsev, "Second-order guarantees of gradient algorithms over networks," in *Proc. Annu. Allerton Conf. Commun., Control, Comput.*, 2018, pp. 359–365.
[35] A. R. Conn, K. Scheinberg, and L. N. Vicente, *Introduction to Derivative-Free Optimization* (MPS-SIAM Series on Optimization). Philadelphia, PA, USA: SIAM, 2009.
[36] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," in *Proc. ACM Workshop Artif. Intell. Secur.*, 2017, pp. 15–26.
[37] Y. Nesterov and V. Spokoiny, "Random gradient-free minimization of convex functions," *Found. Comput. Math.*, vol. 17, no. 2, pp. 527–566, 2017.
[38] A. Agarwal, O. Dekel, and L. Xiao, "Optimal algorithms for online convex optimization with multi-point bandit feedback," in *Proc. Conf. Learn. Theory*, 2010, pp. 28–40.

[39] D. Yuan and D. W. Ho, "Randomized gradient-free method for multiagent optimization over time-varying networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 6, pp. 1342–1347, Jun. 2015.

[40] D. Yuan, S. Xu, and J. Lu, "Gradient-free method for distributed multi-agent optimization via push-sum algorithms," *Int. J. Robust Nonlinear Control*, vol. 25, no. 10, pp. 1569–1580, 2015.

[41] A. K. Sahu, D. Jakovetic, D. Bajovic, and S. Kar, "Distributed zeroth order optimization over random networks: A Kiefer-Wolfowitz stochastic approximation approach," in *Proc. IEEE Conf. Decis. Control*, 2018, pp. 4951–4958.

[42] Y. Wang, W. Zhao, Y. Hong, and M. Zamani, "Distributed subgradient-free stochastic optimization algorithm for nonsmooth convex functions over time-varying networks," *SIAM J. Control Optim.*, vol. 57, no. 4, pp. 2821–2842, 2019.

[43] Y. Pang and G. Hu, "Randomized gradient-free distributed optimization methods for a multi-agent system with unknown cost function," *IEEE Trans. Autom. Control*, vol. 65, no. 1, pp. 333–340, Jan. 2020.

[44] Z. Yu, D. W. Ho, and D. Yuan, "Distributed randomized gradient-free mirror descent algorithm for constrained optimization," *IEEE Trans. Autom. Control*, to be published, doi: 10.1109/TAC.2021.3075669.

[45] D. Hajinezhad, M. Hong, and A. Garcia, "ZONE: Zeroth-order nonconvex multiagent optimization over networks," *IEEE Trans. Autom. Control*, vol. 64, no. 10, pp. 3995–4010, Oct. 2019.

[46] Y. Tang, J. Zhang, and N. Li, "Distributed zero-order algorithms for nonconvex multi-agent optimization," *IEEE Control Netw. Syst.*, vol. 8, no. 1, pp. 269–281, Mar. 2021.

[47] R. Xin, U. A. Khan, and S. Kar, "A fast randomized incremental gradient method for decentralized non-convex optimization," 2020, *arXiv:2011.03853*.

[48] X. Yi, S. Zhang, T. Yang, T. Chai, and K. H. Johansson, "Linear convergence of first- and zeroth-order algorithms for distributed nonconvex optimization," 2019, *arXiv:1912.12110*.

[49] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton, NJ, USA: Princeton Univ. Press, 2010.

[50] D. Jakovetić, D. Bajović, J. Xavier, and J. M. Moura, "Primal-dual methods for large-scale and distributed convex optimization and data analytics," *Proc. IEEE*, vol. 108, no. 11, pp. 1923–1938, Nov. 2020.