Automatica 142 (2022) 110353

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Zeroth-order algorithms for stochastic distributed nonconvex optimization[☆]

Xinlei Yi[a], Shengjun Zhang[b], Tao Yang[c,*], Karl H. Johansson[a]

[a] *School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, and Digital Futures, 10044, Stockholm, Sweden*
[b] *Department of Electrical Engineering, University of North Texas, Denton, TX 76203, USA*
[c] *State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, 110819, Shenyang, China*

ARTICLE INFO

Article history:
Received 14 June 2021
Received in revised form 27 February 2022
Accepted 4 March 2022
Available online 11 May 2022

Keywords:
Distributed nonconvex optimization
Gradient-free
Linear speedup
Polyak-Łojasiewicz condition
Stochastic optimization

ABSTRACT

In this paper, we consider a stochastic distributed nonconvex optimization problem with the cost function being distributed over n agents having access only to zeroth-order (ZO) information of the cost. This problem has various machine learning applications. As a solution, we propose two distributed ZO algorithms, in which at each iteration each agent samples the local stochastic ZO oracle at two points with a time-varying smoothing parameter. We show that the proposed algorithms achieve the linear speedup convergence rate $\mathcal{O}(\sqrt{p/(nT)})$ for smooth cost functions under the state-dependent variance assumptions which are more general than the commonly used bounded variance and Lipschitz assumptions, and $\mathcal{O}(p/(nT))$ convergence rate when the global cost function additionally satisfies the Polyak–Łojasiewicz (P–Ł) condition, where p and T are the dimension of the decision variable and the total number of iterations, respectively. To the best of our knowledge, this is the first linear speedup result for distributed ZO algorithms. It consequently enables systematic processing performance improvements by adding more agents. We also show that the proposed algorithms converge linearly under the relatively bounded second moment assumptions and the P–Ł condition. We demonstrate through numerical experiments the efficiency of our algorithms on generating adversarial examples from deep neural networks in comparison with baseline and recently proposed centralized and distributed ZO algorithms.

1. Introduction

We consider stochastic distributed nonconvex optimization with zeroth-order (ZO) information feedback. Specifically, consider a network of n agents/machines collaborating to solve the following optimization problem

$$\min_{x\in\mathbb{R}^p} f(x) := \frac{1}{n}\sum_{i=1}^{n}\mathbf{E}_{\xi_i}[F_i(x,\xi_i)], \tag{1}$$

where $x \in \mathbb{R}^p$ is the decision variable, ξ_i is a random variable, and $F_i(\cdot,\xi_i) : \mathbb{R}^p \mapsto \mathbb{R}$ is a stochastic component function (not necessarily convex). Each agent i only has information about its own stochastic ZO oracle $F_i(x,\xi_i)$. In other words, for any given x and ξ_i, each agent i can sample $F_i(x,\xi_i)$ as a stochastic approximation of the true local cost function value $f_i(x) = \mathbf{E}_{\xi_i}[F_i(x,\xi_i)]$, but other information such as the first-order oracle cannot be observed. Agents communicate with their neighbors through an underlying communication network. The network is modeled by an undirected graph $\mathcal{G} = (\mathcal{V},\mathcal{E})$, where $\mathcal{V} = \{1,\ldots,n\}$ is the agent set, $\mathcal{E} \subseteq \mathcal{V}\times\mathcal{V}$ the edge set, and $(i,j) \in \mathcal{E}$ if agents i and j communicate with each other. The neighboring set of agent i is denoted by $\mathcal{N}_i = \{j \in \mathcal{V} : (i,j) \in \mathcal{E}\}$. The ZO information feedback setting has wide usage in applications, particularly when explicit expressions of the gradients are unavailable or difficult to obtain (Audet & Hare, 2017; Conn, Scheinberg, & Vicente, 2009b; Larson, Menickelly, & Wild, 2019). For example, the cost functions of many big data problems that deal with complex data generating processes cannot be explicitly defined (Chen, Zhang, Sharma, Yi, & Hsieh, 2017). Moreover, the distributed setting is a core aspect of many important applications in view of flexibility and scalability to large-scale datasets and systems, data privacy and locality, communication reduction to the central entity, and robustness to potential failures of the central entity (Koloskova, Stich, & Jaggi, 2019; Nedić & Liu, 2018; Yang et al., 2019).

[☆] This work was supported by the Knut and Alice Wallenberg Foundation, the Swedish Foundation for Strategic Research, the Swedish Research Council, and the National Natural Science Foundation of China under grants 62133003, 61991403, and 61991400. The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Luca Schenato under the direction of Editor Christos G. Cassandras.

[*] Corresponding author.

E-mail addresses: xinleiy@kth.se (X. Yi), ShengjunZhang@my.unt.edu (S. Zhang), yangtao@mail.neu.edu.cn (T. Yang), kallej@kth.se (K.H. Johansson).

https://doi.org/10.1016/j.automatica.2022.110353
0005-1098/© 2022 Elsevier Ltd. All rights reserved.

1.1. Literature review

The study of gradient-free (derivative-free) optimization has a long history, which can be traced back at least to the 1960s (Hooke & Jeeves, 1961; Matyas, 1965; Nelder & Mead, 1965). It has recently gained renewed attention by the machine learning community. Classical gradient-free optimization methods can be classified into direct-search and model-based methods. For example, stochastic direct-search and model-based trust-region algorithms have been proposed in Bergou, Gorbunov, and Richtárik (2020), Bibi, Bergou, Sener, Ghanem, and Richtárik (2020), Conn, Scheinberg, and Vicente (2009a), Golovin et al. (2020), Gorbunov, Bibi, Sener, Bergou, and Richtárik (2020), Marazzi and Nocedal (2002) and Scheinberg and Toint (2010), respectively. In recent years, the more popular gradient-free optimization methods are ZO methods, which are gradient-free counterparts of first-order optimization methods and thus easy to implement. In ZO optimization methods, the full or stochastic gradients are approximated by directional derivatives, which are calculated through sampled function values. A commonly used method to calculate directional derivatives is simply using the function difference at two points (Duchi, Jordan, Wainwright, & Wibisono, 2015; Nesterov & Spokoiny, 2017; Shamir, 2017).

Various ZO optimization methods have been proposed, e.g., ZO (stochastic) gradient descent algorithms (Bach & Perchet, 2016; Balasubramanian & Ghadimi, 2018; Ghadimi & Lan, 2013; Jin, Liu, Ge, & Jordan, 2018; Kozak, Becker, Doostan, & Tenorio, 2021; Liu, Chen, Chen, & Hong, 2019; Liu, Li, Chen, Haupt, & Amini, 2018; Nesterov & Spokoiny, 2017; Shamir, 2013; Vlatakis-Gkaragkounis, Flokas, & Piliouras, 2019; Ye, Huang, Fang, Li, & Zhang, 2018; Zhang, Zhou, Ji, & Zavlanos, 2022), ZO stochastic coordinate descent algorithms (Lian, Zhang, Hsieh, Huang, & Liu, 2016), ZO (stochastic) variance reduction algorithms (Balasubramanian & Ghadimi, 2018; Chen, Orvieto, & Lucchi, 2020; Fang, Li, Lin, & Zhang, 2018; Gao & Huang, 2020; Gao, Jiang, & Zhang, 2018; Ghadimi, Lan, & Zhang, 2016; Gorbunov, Dvurechensky, & Gasnikov, 2018; Gu, Huo, Deng, & Huang, 2018; Huang, Gu, Huo, Chen, & Huang, 2019; Huang, Tao, & Chen, 2020; Ji, Wang, Zhou, & Liang, 2019; Jin et al., 2018; Kazemi & Wang, 2018; Liu et al., 2019; Liu, Cheng, Hsieh, & Tao, 2018; Liu, Kailkhura, et al., 2018; Liu, Li, Chen, Haupt, & Amini, 2018), ZO (stochastic) proximal algorithms (Cai, Mckenzie, Yin, & Zhang, 2020; Ghadimi et al., 2016; Huang, Gu, Huo, Chen, & Huang, 2019; Nazari, Tarzanagh, & Michailidis, 2020), ZO Frank–Wolfe algorithms (Balasubramanian & Ghadimi, 2018; Gao & Huang, 2020; Huang et al., 2020; Sahu, Zaheer, & Kar, 2019), ZO mirror descent algorithms (Duchi et al., 2015; Gorbunov et al., 2018; Wang, Du, Balakrishnan, & Singh, 2018), ZO adaptive momentum methods (Chen et al., 2019; Nazari et al., 2020), ZO methods of multipliers (Gao et al., 2018; Huang, Gao, Chen, & Huang, 2019; Huang, Gao, Pei, & Huang, 2019; Kazemi & Wang, 2018), ZO stochastic path-integrated differential estimator (Fang et al., 2018; Huang, Gao, Pei, & Huang, 2019; Ji et al., 2019). Convergence properties of these algorithms have been analyzed in detail. For instance, the typical convergence result for deterministic centralized optimization problems is that first-order stationary points can be found at an $\mathcal{O}(p/T)$ convergence rate by the two-point sampling based ZO algorithms (Kozak et al., 2021; Nesterov & Spokoiny, 2017), while under stochastic settings the convergence rate is reduced to $\mathcal{O}(\sqrt{p/T})$ (Ghadimi & Lan, 2013; Lian et al., 2016), where T is the total number of iterations.

Aforementioned ZO optimization algorithms are all centralized and thus not suitable to solve distributed optimization problems. Recently distributed ZO algorithms have been proposed, e.g., distributed ZO gradient descent algorithms (Pang & Hu, 2020; Sahu, Jakovetić, Bajović, & Kar, 2018a, 2018b; Tang, Zhang, & Li, 2020; Wang, Zhao, Hong, & Zamani, 2019; Yuan & Ho, 2014), distributed ZO push-sum algorithm (Yuan, Xu, & Lu, 2015), distributed ZO mirror descent algorithm (Yu, Ho, & Yuan, 2022), distributed ZO gradient tracking algorithm (Tang et al., 2020), distributed ZO primal–dual algorithms (Hajinezhad, Hong, & Garcia, 2019; Hajinezhad & Zavlanos, 2018; Yi, Zhang, Yang, Chai, & Johansson, 2021), distributed ZO sliding algorithm (Beznosikov, Gorbunov, & Gasnikov, 2020), privacy-preserving distributed ZO algorithm (Gratton, Venkategowda, Arablouei, & Werner, 2021), distributed ZO Frank–Wolfe algorithm (Sahu & Kar, 2020). Among these algorithms, the algorithms in Sahu et al. (2018a, 2018b), Tang et al. (2020), Yi, Zhang, Yang, Chai, and Johansson (2021), Yu et al. (2022), Yuan and Ho (2014) and Yuan et al. (2015) are suitable to solve the deterministic form of (1), while the algorithm in Hajinezhad et al. (2019) can be directly applied to solve the stochastic optimization problem (1). However, the algorithm in Hajinezhad et al. (2019) requires each agent to have an $\mathcal{O}(T)$ sampling size per iteration, which is not favorable in practice, although it was shown that first-order stationary points can be found at an $\mathcal{O}(p^2n/T)$ convergence rate.

From the discussions above, three core theoretical questions arise when considering stochastic distributed optimization problems:

(Q1) Can distributed ZO algorithms achieve similar convergence properties as centralized ZO algorithms? For instance, can distributed ZO algorithms based on two-point sampling have an $\mathcal{O}(\sqrt{p/T})$ convergence rate as their centralized counterparts in Ghadimi and Lan (2013) and Lian et al. (2016)?

(Q2) As shown in Lian et al. (2017), distributed stochastic gradient descent (SGD) algorithms can achieve linear speedup with respect to the number of agents compared with centralized SGD algorithms. Can distributed ZO algorithms also achieve linear speedup compared with centralized ZO algorithms? In particular, can distributed ZO algorithms based on two-point sampling achieve the linear speedup convergence rate $\mathcal{O}(\sqrt{p/nT})$?

(Q3) For deterministic optimization problems, centralized and distributed ZO algorithms can achieve faster convergence rates under more stringent conditions such as strong convexity or Polyak–Łojasiewicz (P–Ł) conditions, as shown in Cai et al. (2020), Chen et al. (2020), Ji et al. (2019), Kozak et al. (2021), Nesterov and Spokoiny (2017), Tang et al. (2020), Ye et al. (2018) and Yi, Zhang, Yang, Chai, and Johansson (2021), respectively. For stochastic optimization problems, can ZO algorithms also achieve faster convergence rates under strong convexity or P–Ł conditions?

1.2. Main contributions

This paper provides positive answers to the above three questions. We propose two distributed ZO algorithms, one primal–dual and one primal algorithm, to solve the stochastic optimization problem (1). In both algorithms, at each iteration each agent communicates its local primal variables to its neighbors through an arbitrarily connected communication network. Moreover, each agent samples its local stochastic ZO oracle at two points with a time-varying smoothing parameter. The contributions of this paper are summarized as follows.

(C1) We show in Theorem 2 that our algorithms find a stationary point with the linear speedup convergence rate $\mathcal{O}(\sqrt{p/(nT)})$ for nonconvex but smooth cost functions under the state-dependent variance assumptions, which are more general than the commonly used bounded variance and Lipschitz assumptions. This rate is faster than that achieved by the centralized ZO algorithms in Balasubramanian and Ghadimi (2018), Chen et al. (2019), Ghadimi and Lan (2013), Lian et al. (2016), Liu et al. (2019), Liu, Li, Chen, Haupt, and Amini (2018) and Zhang et al. (2022) and the distributed ZO algorithm in Tang et al. (2020). To the best of our knowledge, this is the first linear speedup result for distributed ZO algorithms; thus (Q1) and (Q2) are answered.

(C2) We show in Theorems 4 and 5 that our proposed algorithms find a global optimum with an $\mathcal{O}(p/(nT))$ convergence rate when the global cost function satisfies the P–Ł condition in addition. This rate is faster than that achieved by the centralized ZO algorithms in Bach and Perchet (2016) and Shamir (2013) and the distributed ZO algorithms in Sahu et al. (2018b) and Tang et al. (2020), although Bach and Perchet (2016), Sahu et al. (2018b) and Shamir (2013) assumed strongly convex cost functions and only considered additive sampling noise, and Tang et al. (2020) only considered the deterministic problem. This paper presents the first performance analysis for ZO algorithms to solve stochastic optimization problems under P–Ł or strong convexity assumptions; thus (Q3) is answered.

(C3) We show in Theorem 6 that our algorithms with constant algorithm parameters linearly converge to a neighborhood of a global optimum under the P–Ł condition. Moreover, an exact global optimum can be linearly found if the relatively bounded second moment assumptions also hold, see Corollary 1. It should be mentioned that the P–Ł constant is not used to design the algorithm parameters when showing these results. Compared with Cai et al. (2020), Chen et al. (2020), Ji et al. (2019), Kozak et al. (2021), Nesterov and Spokoiny (2017) and Ye et al. (2018) which also achieved linear convergence, we use less restrictive assumptions on the cost function and/or less samplings per iteration.

The detailed comparison of this paper to other related studies in the literature in terms of problem settings, number of sampled points per iteration, convergence rate, and sampling complexity is summarized in a table provided in the online version (Yi, Zhang, Yang, & Johansson, 2021) due to the space limitation.

1.3. Outline

The rest of this paper is organized as follows. Section 2 introduces some preliminaries. Sections 3 and 4 provide the distributed primal–dual and primal ZO algorithms, respectively, and analyze their convergence properties. Numerical evaluations for generating adversarial examples from black-box deep neural networks are given in Section 5. Finally, concluding remarks are offered in Section 6. All the proofs are given in the online version (Yi, Zhang, Yang, & Johansson, 2021).

Notations. $\mathbb{N}_0$ and $\mathbb{N}_+$ denote the set of nonnegative and positive integers, respectively. $[n]$ denotes the set $\{1,\dots,n\}$ for any $n\in\mathbb{N}_+$. $\|\cdot\|$ represents the Euclidean norm for vectors or the induced 2-norm for matrices. $\mathbb{B}^p$ and $\mathbb{S}^p$ are the unit ball and sphere centered around the origin in $\mathbb{R}^p$ under Euclidean norm, respectively. Given a differentiable function f, ∇f denotes its gradient.

2. Preliminaries

In this section, we introduce the P–Ł condition, the random gradient estimator, and the assumptions used in this paper.

2.1. Polyak–Łojasiewicz condition

Definition 1 (*Karimi, Nutini, & Schmidt, 2016*)**.** A differentiable function $f(x):\ \mathbb{R}^p\mapsto\mathbb{R}$ satisfies the Polyak–Łojasiewicz (P–Ł) condition with constant $\nu>0$ if $f^*>-\infty$, where $f^*=\min_{x\in\mathbb{R}^p}f(x)$, and

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

It is straightforward to see that every (essentially, weakly, or restricted) strongly convex function satisfies the P–Ł condition. The P–Ł condition implies that every stationary point is a global minimizer. But unlike (essentially, weakly, or restricted) strong convexity, the P–Ł condition alone does not imply convexity of f. Moreover, it does not imply that the set of global minimizers is a singleton (Karimi et al., 2016; Zhang & Cheng, 2015). In fact, P–Ł condition generalizes strong convexity to nonconvex functions. The function $f(x)=x^2+3\sin^2(x)$ given in Karimi et al. (2016) is an example of a nonconvex function satisfying the P–Ł condition with $\nu=1/32$. Moreover, it was shown in Li and Li (2018) that the loss functions in some applications satisfy the P–Ł condition in the region near a local minimum. Moreover, Fazel, Ge, Kakade, and Mesbahi (2018) proved that the cost function of the policy optimization for the linear quadratic regulator problem is nonconvex but satisfies the P–Ł condition. More examples of nonconvex functions satisfying the P–Ł condition can be found in Karimi et al. (2016) and Zhang and Cheng (2015).

2.2. Gradient estimator

Let $f(x):\ \mathbb{R}^p\mapsto\mathbb{R}$ be a function. Duchi et al. (2015) proposed the following random gradient estimator:

$$\hat{\nabla}_2 f(x,\delta,u)=\frac{p}{\delta}(f(x+\delta u)-f(x))u, \tag{3}$$

where $\delta>0$ is the smoothing/exploration parameter and $u\in\mathbb{S}^p$ is a uniformly distributed random vector. This gradient estimator can be calculated by sampling the function f at two points (e.g., x and $x+\delta u$). The intuition of this estimator follows from directional derivatives (Duchi et al., 2015). From a practical point of view, the larger the smoothing parameter δ the better, since in this case it is easier to distinguish the two sampled function values.

2.3. Assumptions

The following assumptions are made.

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

Assumption 2. The optimal set $\mathbb{X}^*$ is nonempty and $f^*>-\infty$, where $\mathbb{X}^*$ and f^* are the optimal set and the minimum function value of the optimization problem (1), respectively.

Assumption 3. For almost all ξ_i, the stochastic ZO oracle $F_i(\cdot,\xi_i)$ is smooth with constant $L_f>0$.

Assumption 4. Each stochastic gradient $\nabla_x F_i(x,\xi_i)$ has state-dependent variance, i.e., there exist two constants σ_0 and σ_1 such that $\mathbf{E}_{\xi_i}[\|\nabla_x F_i(x,\xi_i)-\nabla f_i(x)\|^2]\leq\sigma_0^2\|\nabla f_i(x)\|^2+\sigma_1^2,\ \forall i\in[n],\ \forall x\in\mathbb{R}^p$.

Assumption 5. Each local gradient $\nabla f_i(x)$ has state-dependent variance, i.e., there exist two constants $\tilde{\sigma}_0$ and σ_2 such that $\|\nabla f_i(x)-\nabla f(x)\|^2\leq\tilde{\sigma}_0^2\|\nabla f(x)\|^2+\sigma_2^2,\ \forall i\in[n],\ \forall x\in\mathbb{R}^p$. Here $\nabla f_i(x)$ can be viewed as a stochastic gradient of $\nabla f(x)$ by randomly picking an index $i\in[n]$.

Assumption 6. The global cost function $f(x)$ satisfies the P–Ł condition with constant $\nu>0$.

Remark 1. It should be highlighted that no convexity assumptions are made. Assumption 1 is common in distributed optimization, e.g., Beznosikov et al. (2020), Hajinezhad et al. (2019), Nedić, Olshevsky, Shi, and Uribe (2017), Qu and Li (2018, 2020), Shi, Ling, Wu, and Yin (2015) and Tang et al. (2020). Assumption 2 is basic. Assumption 3 is standard in stochastic optimization with ZO information feedback, e.g., Balasubramanian and Ghadimi (2018), Gao et al. (2018), Ghadimi and Lan (2013), Ghadimi et al. (2016), Gorbunov et al. (2018), Hajinezhad et al. (2019), Kazemi and Wang (2018), Lian et al. (2016) and Liu, Cheng, Hsieh, and Tao (2018). When $\sigma_0 = 0$, Assumption 4 recovers the bounded variance assumption, which is commonly used in the literature studying stochastic ZO optimization, e.g., Balasubramanian and Ghadimi (2018), Gao et al. (2018), Ghadimi and Lan (2013), Ghadimi et al. (2016), Gorbunov et al. (2018), Hajinezhad et al. (2019), Kazemi and Wang (2018), Lian et al. (2016) and Sahu et al. (2019). Therefore, Assumption 4 is more general. When $\tilde{\sigma}_0 = 0$, Assumption 5 becomes the bounded variance assumption, i.e., $\|\nabla f_i(x) - \nabla f(x)\|^2$ is globally bounded, and is weaker than the Lipschitz assumption, i.e., $\|\nabla f_i(x)\|$ is globally bounded. Both the bounded variance and Lipschitz assumptions are normally used in the literature studying ZO optimization, e.g., Duchi et al. (2015), Fang et al. (2018), Gao et al. (2018), Gu et al. (2018), Hajinezhad et al. (2019), Huang, Gao, Chen, and Huang (2019), Huang, Gao, Pei, and Huang (2019), Huang, Gu, Huo, Chen, and Huang (2019), Ji et al. (2019), Kazemi and Wang (2018), Liu et al. (2019), Liu, Kailkhura, et al. (2018), Liu, Li, Chen, Haupt, and Amini (2018), Tang et al. (2020), Yuan and Ho (2014) and Yuan et al. (2015), respectively. However, the Lipschitz assumption is too restrictive since even a simple quadratic function is typically not Lipschitz. Moreover, the bounded variance assumption is also restrictive, for instance it is impractical to assume this assumption for distributed learning problems with local cost functions being constructed by heterogeneous data collected locally by agents. In contrast, Assumption 5 is more general due to the state-dependent term $\tilde{\sigma}_0^2\|\nabla f(x)\|^2$, and it is not needed when Assumption 6 holds and the constant ν is known in advance as shown in Theorem 5. Assumption 6 is weaker than the assumption that the global or each local cost function is (restricted) strongly convex. It plays a key role to guarantee that a global optimum can be found and to show that faster convergence rate can be achieved.

To end this section, we introduce the stronger alternatives of Assumptions 4 and 5, which are used to show faster convergence for the proposed algorithms.

Assumption 4′. The second moment of each stochastic gradient $\nabla_x F_i(x,\xi_i)$ is relatively bounded, i.e., there exists a constant $\breve{\sigma}_0$ such that $\mathbf{E}_{\xi_i}[\|\nabla_x F_i(x,\xi_i)\|^2] \le \breve{\sigma}_0^2\|\nabla f_i(x)\|^2,\ \forall i\in[n],\ \forall x\in\mathbb{R}^p$.

Assumption 5′. The second moment of each local gradient $\nabla f_i(x)$ is relatively bounded, i.e., there exists a constant $\hat{\sigma}_0$ such that $\|\nabla f_i(x)\|^2 \le \hat{\sigma}_0^2\|\nabla f(x)\|^2,\ \forall i\in[n],\ \forall x\in\mathbb{R}^p$.

It is straightforward to check that Assumption 4′ (Assumption 5′) is equivalent to Assumption 4 (Assumption 5) when $\sigma_1 = 0$ ($\sigma_2 = 0$). Assumption 4′ is satisfied trivially when the deterministic ZO information is available. Assumption 5′ holds when $\nabla f_i(x)$ is proportional to $\nabla f(x)$, for example when all the random variables ξ_i have a common probability distribution and the local stochastic component functions are the same, which is a common setup in distributed empirical risk minimization problems. Moreover, for deterministic centralized optimization problems, Assumptions 4′ and 5′ hold trivially.

3. Distributed ZO primal–dual algorithm

In this section, we propose a distributed ZO primal–dual algorithm and analyze its convergence properties.

When gradient information is available, in Yi, Zhang, Yang, Chai, and Johansson (2021) the following distributed first-order primal–dual algorithm was proposed to solve (1):

$$x_{i,k+1} = x_{i,k} - \eta\Big(\alpha\sum_{j\in\mathcal{N}_i}L_{ij}x_{j,k} + \beta v_{i,k} + \nabla f_i(x_{i,k})\Big), \tag{4a}$$

$$v_{i,k+1} = v_{i,k} + \eta\beta\sum_{j\in\mathcal{N}_i}L_{ij}x_{j,k},$$

$$\forall x_{i,0}\in\mathbb{R}^p,\ \sum_{j=1}^n v_{j,0} = \mathbf{0}_p,\ \forall i\in[n], \tag{4b}$$

where α, β, and η are positive algorithm parameters, $\mathcal{N}_i$ is the neighboring set of agent i as defined below (1), and $L = [L_{ij}]$ is the weighted Laplacian matrix associated with the undirected communication graph $\mathcal{G}$. As pointed out in Yi, Zhang, Yang, Chai, and Johansson (2021), the distributed first-order algorithm (4) is a special form of several existing first-order algorithms in the literature, e.g., Jakovetić, Bajović, Xavier, and Moura (2020) and Shi et al. (2015), and it has been shown that this algorithm can find a stationary point with an $\mathcal{O}(1/k)$ convergence rate.

Noting that we consider the scenario where only stochastic ZO oracles rather than the explicit expressions of the gradients are available, we need to estimate the gradients used in the distributed first-order algorithm (4). Inspired by (3), we introduce

$$g^e_{i,k} = \frac{p(F_i(x_{i,k}+\delta_{i,k}u_{i,k},\xi_{i,k}) - F_i(x_{i,k},\xi_{i,k}))}{\delta_{i,k}}u_{i,k}, \tag{5}$$

where $\delta_{i,k} > 0$ is a time-varying smoothing parameter and $u_{i,k}\in\mathbb{S}^p$ is a uniformly distributed random vector chosen by agent i at iteration k; $\xi_{i,k}$ is a random variable sampled by agent i at iteration k according to the distribution of ξ_i; and $F_i(x_{i,k}+\delta_{i,k}u_{i,k},\xi_{i,k})$ and $F_i(x_{i,k},\xi_{i,k})$ are the values sampled by agent i at iteration k. We replace the gradient and fixed algorithm parameters in (4) with the stochastic gradient estimator (5) and time-varying parameters, respectively. Then we get the following ZO algorithm:

$$x_{i,k+1} = x_{i,k} - \eta_k\Big(\alpha_k\sum_{j\in\mathcal{N}_i}L_{ij}x_{j,k} + \beta_k v_{i,k} + g^e_{i,k}\Big), \tag{6a}$$

$$v_{i,k+1} = v_{i,k} + \eta_k\beta_k\sum_{j\in\mathcal{N}_i}L_{ij}x_{j,k},$$

$$\forall x_{i,0}\in\mathbb{R}^p,\ \sum_{j=1}^n v_{j,0} = \mathbf{0}_p,\ \forall i\in[n]. \tag{6b}$$

We write the distributed ZO algorithm (6) in pseudo-code as Algorithm 1. In this algorithm, from the way to generate $u_{i,k}$ and $\xi_{i,k}$, we know that $u_{i,k},\ \xi_{j,l},\ \forall i,j\in[n],\ k,l\in\mathbb{N}_+$ are mutually independent. Let $\mathfrak{L}_k$ denote the σ-algebra generated by the independent random variables $u_{1,k},\dots,u_{n,k},\xi_{1,k},\dots,\xi_{n,k}$ and let $\mathcal{L}_k = \bigcup_{t=0}^k\mathfrak{L}_t$. From the independence property of $u_{i,k}$ and $\xi_{i,l}$, we can see that $x_{i,k}$ and $v_{i,k+1},\ i\in[n]$ depend on $\mathcal{L}_{k-1}$ and are independent of $\mathfrak{L}_t$ for all $t\ge k$.

Remark 2. In Algorithm 1, each agent i maintains two local sequences, i.e., the local primal and dual variable sequences $\{x_{i,k}\}$ and $\{v_{i,k}\}$, and communicates its local primal variables to its neighbors through the network. Moreover, at each iteration each agent samples its local stochastic ZO oracle at two points to estimate the gradient of its local cost function. It should be

Algorithm 1 Distributed ZO Primal–Dual Algorithm

```
Input: positive sequences $\{\alpha_k\}$, $\{\beta_k\}$, $\{\eta_k\}$, and $\{\delta_{i,k}\}$.
Initialize: $x_{i,0} \in \mathbb{R}^p$ and $v_{i,0} = \mathbf{0}_p$, $\forall i \in [n]$.
for $k = 0, 1, \dots$ do
    for $i = 1, \dots, n$ in parallel do
        Broadcast $x_{i,k}$ to $\mathcal{N}_i$ and receive $x_{j,k}$ from $j \in \mathcal{N}_i$;
        Generate $u_{i,k} \in \mathbb{S}^p$ independently and uniformly at random;
        Generate $\xi_{i,k}$ independently and randomly according to the distribution of $\xi_i$;
        Sample $F_i(x_{i,k}, \xi_{i,k})$ and $F_i(x_{i,k} + \delta_{i,k}u_{i,k}, \xi_{i,k})$;
        Update $x_{i,k+1}$ by (6a);
        Update $v_{i,k+1}$ by (6b).
    end for
end for
Output: $\{\boldsymbol{x}_k\}$.
```

highlighted that the agent-wise smoothing parameter $\delta_{i,k}$ is time-varying. It can in many situations be chosen larger than the fixed smoothing parameter used in existing ZO algorithms. For example, in the following we use an $\mathcal{O}(1/k^{1/4})$ smoothing parameter, which is larger than the $\mathcal{O}(1/T^{1/2})$ smoothing parameter used in Ghadimi and Lan (2013).

3.1. Find stationary points

Let us consider the case when Algorithm 1 is able to find stationary points. We first have the following convergence result.

Theorem 1. *Suppose Assumptions 1–5 hold. Let $\{\boldsymbol{x}_k\}$ be the sequence generated by Algorithm 1 with*

$$\alpha_k = \kappa_1\beta_k,\ \ \beta_k = \kappa_0(k+t_1)^{\theta},\ \ \eta_k = \frac{\kappa_2}{\beta_k},$$

$$\delta_{i,k} \le \frac{\kappa_\delta\sqrt{p\eta_k}}{\sqrt{n+p}},\ \ \forall k \in \mathbb{N}_0, \tag{7}$$

where $\kappa_1 > c_1$, $\kappa_2 \in (0, c_2(\kappa_1))$, $\theta \in (0.5, 1)$, $t_1 \ge (\sqrt{p}c_3(\kappa_1, \kappa_2))^{1/\theta}$, $\kappa_0 \ge c_0(\kappa_1, \kappa_2)/t_1^{\theta}$, and $\kappa_\delta > 0$ with the explicit expressions of $c_0(\kappa_1, \kappa_2)$, c_1, $c_2(\kappa_1)$, and $c_3(\kappa_1, \kappa_2)$ being given in Yi, Zhang, Yang, and Johansson (2021). Then, for any $T \in \mathbb{N}_+$,

$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbf{E}[\|\nabla f(\bar{x}_k)\|^2] = \mathcal{O}\Big(\frac{\sqrt{p}}{T^{1-\theta}} + \frac{p}{T}\Big), \tag{8a}$$

$$\mathbf{E}[f(\bar{x}_T)] - f^* = \mathcal{O}(1), \tag{8b}$$

$$\mathbf{E}\Big[\frac{1}{n}\sum_{i=1}^{n}\|x_{i,T} - \bar{x}_T\|^2\Big] = \mathcal{O}\Big(\frac{1}{T^{2\theta}}\Big), \tag{8c}$$

where $\bar{x}_k = \frac{1}{n}\sum_{i=1}^{n} x_{i,k}$.

Proof. The explicit expressions of the right-hand sides of (8a)–(8c) and the proof are given in Yi, Zhang, Yang, and Johansson (2021). □

If the total number of iterations T and the number of agents n are known in advance, then, as shown in the following, Algorithm 1 can find a stationary point of (1) with an $\mathcal{O}(\sqrt{p/(nT)})$ convergence rate, and thus achieves linear speedup with respect to the number of agents compared to the $\mathcal{O}(\sqrt{p/T})$ convergence rate achieved by the centralized stochastic ZO algorithms in Ghadimi and Lan (2013) and Lian et al. (2016). The linear speedup property enables us to scale up the computing capability by adding more agents into the algorithm (Yu, Jin, & Yang, 2019).

Theorem 2 (*Linear Speedup*). *Suppose Assumptions 1–5 hold. For any given $T > \max\{n(\tilde{c}_0(\kappa_1, \kappa_2)/\kappa_2)^2,\ n^3\}/p$, let $\{\boldsymbol{x}_k, k = 0, \dots, T\}$ be the output generated by Algorithm 1 with*

$$\alpha_k = \kappa_1\beta_k,\ \ \beta_k = \beta = \frac{\kappa_2\sqrt{pT}}{\sqrt{n}},\ \ \eta_k = \frac{\kappa_2}{\beta_k},$$

$$\delta_{i,k} \le \frac{p^{1/4}n^{1/4}\kappa_\delta}{\sqrt{n+p}(k+1)^{1/4}},\ \ \forall k \le T, \tag{9}$$

where $\kappa_1 > c_1$, $\kappa_2 \in (0, c_2(\kappa_1))$, and $\kappa_\delta > 0$ with the explicit expressions of $\tilde{c}_0(\kappa_1, \kappa_2)$, c_1, and $c_2(\kappa_1)$ being given in Yi, Zhang, Yang, and Johansson (2021). Then,

$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbf{E}[\|\nabla f(\bar{x}_k)\|^2] = \mathcal{O}\Big(\frac{\sqrt{p}}{\sqrt{nT}}\Big) + \mathcal{O}\Big(\frac{n}{T}\Big), \tag{10a}$$

$$\mathbf{E}[f(\bar{x}_T)] - f^* = \mathcal{O}(1), \tag{10b}$$

$$\mathbf{E}\Big[\frac{1}{n}\sum_{i=1}^{n}\|x_{i,T} - \bar{x}_T\|^2\Big] = \mathcal{O}\Big(\frac{n}{T}\Big). \tag{10c}$$

Proof. The explicit expressions of the right-hand sides of (10a)–(10c) and the proof are given in Yi, Zhang, Yang, and Johansson (2021). It should be highlighted that the omitted constants in the first term on the right-hand side of (10a) do not depend on any parameters related to the communication network. □

Remark 3. To the best of our knowledge, Theorem 2 is the first result to establish linear speedup for a distributed ZO algorithm to solve stochastic optimization problems. The achieved rate is faster than that achieved by the centralized ZO algorithms in Balasubramanian and Ghadimi (2018), Chen et al. (2019), Ghadimi and Lan (2013), Lian et al. (2016), Liu et al. (2019), Liu, Li, Chen, Haupt, and Amini (2018) and Zhang et al. (2022) and the distributed ZO gradient descent algorithm in Tang et al. (2020). The rate is slower than that achieved by the centralized ZO algorithms in Fang et al. (2018), Ghadimi et al. (2016), Gu et al. (2018), Huang, Gu, Huo, Chen, and Huang (2019), Ji et al. (2019), Kazemi and Wang (2018), Liu, Cheng, Hsieh, and Tao (2018) and Liu, Kailkhura, et al. (2018), which is reasonable since these algorithms not only are centralized but also use variance reduction techniques. The distributed ZO gradient tracking algorithm in Tang et al. (2020) and the distributed ZO primal–dual algorithms in Hajinezhad et al. (2019) and Yi, Zhang, Yang, Chai, and Johansson (2021) also achieved faster convergence rates than ours. However, in Fang et al. (2018), Gu et al. (2018), Huang, Gu, Huo, Chen, and Huang (2019), Ji et al. (2019), Liu, Kailkhura, et al. (2018), Tang et al. (2020) and Yi, Zhang, Yang, Chai, and Johansson (2021), the considered problems are deterministic; in Tang et al. (2020) and Yi, Zhang, Yang, Chai, and Johansson (2021), the sampling size of each agent at each iteration is $\mathcal{O}(p)$, which results in a heavy sampling burden when p is large; in Ghadimi et al. (2016), Hajinezhad et al. (2019) and Kazemi and Wang (2018), the sampling size of each agent at each iteration is $\mathcal{O}(T)$, which is difficult to execute in practice. One future research direction is to establish faster convergence with reduced sampling complexity by using variance reduction techniques.

3.2. Find global optimum

Let us next consider cases when Algorithm 1 finds global optimum.

Theorem 3. *Suppose Assumptions 1–6 hold. Let $\{\boldsymbol{x}_k\}$ be the sequence generated by Algorithm 1 with*

$$\alpha_k = \kappa_1\beta_k,\ \ \beta_k = \kappa_0(k+t_1)^{\theta},\ \ \eta_k = \frac{\kappa_2}{\beta_k},$$

$$\delta_{i,k} \le \frac{\kappa_\delta \sqrt{p\eta_k}}{\sqrt{n+p}}, \ \ \forall k \in \mathbb{N}_0, \tag{11}$$

where $\kappa_1 > c_1$, $\kappa_2 \in (0, c_2(\kappa_1))$, $\theta \in (0,1)$, $t_1 \in [(pc_3(\kappa_1,\kappa_2))^{1/\theta}, (pc_4c_3(\kappa_1,\kappa_2))^{1/\theta}]$, $\kappa_0 \ge c_0(\kappa_1,\kappa_2)/t_1^\theta$, $\kappa_\delta > 0$, and $c_4 \ge 1$ with the explicit expressions of $c_0(\kappa_1,\kappa_2)$, c_1, $c_2(\kappa_1)$, and $c_3(\kappa_1,\kappa_2)$ being given in Yi, Zhang, Yang, and Johansson (2021). Then, for any $T \in \mathbb{N}_+$,

$$\mathbf{E}\Big[\frac{1}{n}\sum_{i=1}^{n}\|x_{i,T}-\bar{x}_T\|^2\Big] = \mathcal{O}\Big(\frac{p}{T^{2\theta}}\Big), \tag{12a}$$

$$\mathbf{E}[f(\bar{x}_T)-f^*] = \mathcal{O}\Big(\frac{p}{nT^{\theta}}\Big) + \mathcal{O}\Big(\frac{p}{T^{2\theta}}\Big). \tag{12b}$$

Proof. The explicit expressions of the right-hand sides of (12a) and (12b), and the proof are given in Yi, Zhang, Yang, and Johansson (2021). It should be highlighted that the omitted constants in the first term in the right-hand side of (12b) do not depend on any parameters related to the communication network. □

From Theorem 3, we see that the convergence rate is strictly slower than $\mathcal{O}(p/(nT))$. In the following we show that the $\mathcal{O}(p/(nT))$ convergence rate can be achieved if the P–Ł constant ν is known in advance. However, information about the total number of iterations T is not needed.

Theorem 4 (*Linear Speedup*). *Suppose Assumptions 1–6 hold and the P–Ł constant ν is known in advance. Let $\{\boldsymbol{x}_k\}$ be the sequence generated by Algorithm 1 with*

$$\alpha_k = \kappa_1\beta_k, \ \ \beta_k = \kappa_0(k+t_1), \ \ \eta_k = \frac{\kappa_2}{\beta_k},$$

$$\delta_{i,k} \le \frac{\kappa_\delta \sqrt{p\eta_k}}{\sqrt{n+p}}, \ \ \forall k \in \mathbb{N}_0, \tag{13}$$

where $\kappa_1 > c_1$, $\kappa_2 \in (0, c_2(\kappa_1))$, $\kappa_0 \in [3\hat{c}_0\nu\kappa_2/16, 3\nu\kappa_2/16)$, $t_1 > \hat{c}_3(\kappa_0,\kappa_1,\kappa_2)$, $\kappa_\delta > 0$, and $\hat{c}_0 \in (0,1)$ with the explicit expressions of c_1, $c_2(\kappa_1)$, and $\hat{c}_3(\kappa_0,\kappa_1,\kappa_2)$ being given in Yi, Zhang, Yang, and Johansson (2021). Then, for any $T \in \mathbb{N}_+$,

$$\mathbf{E}\Big[\frac{1}{n}\sum_{i=1}^{n}\|x_{i,T}-\bar{x}_T\|^2\Big] = \mathcal{O}\Big(\frac{p}{T^{2}}\Big), \tag{14a}$$

$$\mathbf{E}[f(\bar{x}_T)-f^*] = \mathcal{O}\Big(\frac{p}{nT}\Big) + \mathcal{O}\Big(\frac{p}{T^{2}}\Big). \tag{14b}$$

Proof. The explicit expressions of the right-hand sides of (14a) and (14b), and the proof are given in Yi, Zhang, Yang, and Johansson (2021). It should be highlighted that the omitted constants in the first term in the right-hand side of (14b) do not depend on any parameters related to the communication network. □

Although Assumption 5 is weaker than the bounded gradient assumption, it can be further relaxed by a mild assumption. Specifically, if each $f_i^* > -\infty$, where $f_i^* = \min_{x\in\mathbb{R}^p} f_i(x)$, then without Assumption 5, the convergence results stated in (14a) and (14b) still hold, as shown in the following.

Theorem 5 (*Linear Speedup*). *Suppose Assumptions 1–4 and 6 hold, and the P–Ł constant ν is known in advance, and each $f_i^* > -\infty$. Let $\{\boldsymbol{x}_k\}$ be the sequence generated by Algorithm 1 with*

$$\alpha_k = \kappa_1\beta_k, \ \ \beta_k = \kappa_0(k+t_1), \ \ \eta_k = \frac{\kappa_2}{\beta_k},$$

$$\delta_{i,k} \le \frac{\kappa_\delta \sqrt{p\eta_k}}{\sqrt{n+p}}, \ \ \forall k \in \mathbb{N}_0, \tag{15}$$

where $\kappa_1 > c_1$, $\kappa_2 \in (0, c_2(\kappa_1))$, $\kappa_0 \in [3\hat{c}_0\nu\kappa_2/16, 3\nu\kappa_2/16)$, $t_1 > \check{c}_3(\kappa_0,\kappa_1,\kappa_2)$, $\kappa_\delta > 0$, and $\hat{c}_0 \in (0,1)$ with the explicit expressions of c_1, $c_2(\kappa_1)$, and $\check{c}_3(\kappa_0,\kappa_1,\kappa_2)$ being given in Yi, Zhang, Yang, and Johansson (2021). Then, for any $T \in \mathbb{N}_+$,

$$\mathbf{E}\Big[\frac{1}{n}\sum_{i=1}^{n}\|x_{i,T}-\bar{x}_T\|^2\Big] = \mathcal{O}\Big(\frac{p}{T^{2}}\Big), \tag{16a}$$

$$\mathbf{E}[f(\bar{x}_T)-f^*] = \mathcal{O}\Big(\frac{p}{nT}\Big) + \mathcal{O}\Big(\frac{p}{T^{2}}\Big). \tag{16b}$$

Proof. The explicit expressions of the right-hand sides of (16a) and (16b), and the proof are given in Yi, Zhang, Yang, and Johansson (2021). It should be highlighted that the omitted constants in the first term in the right-hand side of (16b) do not depend on any parameters related to the communication network. □

Remark 4. To the best of our knowledge, Theorems 3–5 are the first performance analysis results for ZO algorithms to solve stochastic optimization problems under the P–Ł condition or strong convexity assumption. In Shamir (2013), a centralized ZO algorithm based on one-point sampling with additive sampling noise was proposed and an $\mathcal{O}(p^2/T)$ convergence rate was achieved for deterministic optimization problems with strongly convex quadratic cost functions. In Bach and Perchet (2016), a centralized ZO algorithm based on two-point sampling with additive noise was proposed and an $\mathcal{O}(p/\sqrt{T})$ convergence rate was achieved for deterministic strongly convex and smooth optimization problems. In Sahu et al. (2018b), a distributed ZO gradient descent algorithm based on $2p$-point sampling with additive noise was proposed and an $\mathcal{O}(pn^2/\sqrt{T})$ convergence rate was achieved for deterministic strongly convex and smooth optimization problems. In Tang et al. (2020), a distributed ZO gradient descent algorithm based on two-point sampling was proposed and an $\mathcal{O}(p/T)$ convergence rate was achieved for deterministic smooth optimization problems under the P–Ł condition. It is straightforward to see that aforementioned convergence rates achieved in Bach and Perchet (2016), Sahu et al. (2018b), Shamir (2013) and Tang et al. (2020) are slower than that achieved by our distributed stochastic ZO primal–dual algorithm as stated in Theorem 5. Moreover, we consider stochastic optimization problems and use the P–Ł condition, which is slightly weaker than the strong convexity condition. The distributed ZO gradient tracking algorithm in Tang et al. (2020) and the distributed ZO primal–dual algorithms in Yi, Zhang, Yang, Chai, and Johansson (2021) achieved linear convergence under the P–Ł condition. However, both algorithms require each agent at each iteration to sample $\mathcal{O}(p)$ points, which results in a heavy sampling burden when p is large.

As shown in Theorems 3–5, in expectation, the convergence rate of Algorithm 1 with diminishing stepsizes is sublinear. The following theorem establishes that, in expectation, the output of Algorithm 1 with constant algorithm parameters linearly converges to a neighborhood of a global optimum.

Theorem 6. *Suppose Assumptions 1–5 hold. Let $\{\boldsymbol{x}_k\}$ be the sequence generated by Algorithm 1 with*

$$\alpha_k = \alpha = \kappa_1\beta, \ \ \beta_k = \beta, \ \ \eta_k = \eta = \frac{\kappa_2}{\beta},$$

$$\delta_{i,k} \le \kappa_\delta \tilde{\varepsilon}^k, \ \ \forall k \in \mathbb{N}_0, \tag{17}$$

where $\kappa_1 > c_1$, $\kappa_2 \in (0, c_2(\kappa_1))$, $\beta \ge \tilde{c}_0(\kappa_1,\kappa_2)$, $\tilde{\varepsilon} \in (0,1)$, and $\kappa_\delta > 0$ with the explicit expressions of $\tilde{c}_0(\kappa_1,\kappa_2)$, c_1, and $c_2(\kappa_1)$ being given in Yi, Zhang, Yang, and Johansson (2021). Then, for any

$T \in \mathbb{N}_+$,

$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbf{E}\Big[\frac{1}{n}\sum_{i=1}^{n}\|x_{i,k}-\bar{x}_k\|^2\Big]$$
$$=\mathcal{O}\Big(\frac{1}{T}+(\sigma_1^2+2(1+\sigma_0^2)\sigma_2^2)p\eta^2\Big), \tag{18a}$$
$$\mathbf{E}\Big[\frac{1}{n}\sum_{i=1}^{n}\|x_{i,T}-\bar{x}_T\|^2\Big]$$
$$=\mathcal{O}\Big(p\eta^2+(\sigma_1^2+2(1+\sigma_0^2)\sigma_2^2)p^2\eta^4\Big(\frac{1}{n}+\eta\Big)T\Big), \tag{18b}$$
$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbf{E}[\|\nabla f(\bar{x}_k)\|^2]$$
$$=\mathcal{O}\Big(\frac{1}{\eta T}+(\sigma_1^2+2(1+\sigma_0^2)\sigma_2^2)\Big(\frac{p\eta}{n}+p\eta^2\Big)\Big). \tag{18c}$$

Moreover, if Assumption 6 also holds, then

$$\mathbf{E}\Big[\frac{1}{n}\sum_{i=1}^{n}\|x_{i,k}-\bar{x}_k\|^2+f(\bar{x}_k)-f^*\Big]$$
$$=\mathcal{O}(\varepsilon^k+(\sigma_1^2+2(1+\sigma_0^2)\sigma_2^2)p\eta),\ \forall k\in\mathbb{N}_+, \tag{19}$$

where $\varepsilon \in (0,1)$ is a constant given in Yi, Zhang, Yang, and Johansson (2021).

Proof. The proof is given in Yi, Zhang, Yang, and Johansson (2021). □

If Assumptions 4′–5′ hold, then $\sigma_1=\sigma_2=0$. In this case, from Theorem 6, we have the following results.

Corollary 1 (*Linear Convergence*)**.** *Under the same setup as Theorem 6 and suppose Assumptions 4′–5′ hold, then, for any $T \in \mathbb{N}_+$,*

$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbf{E}\Big[\frac{1}{n}\sum_{i=1}^{n}\|x_{i,k}-\bar{x}_k\|^2\Big]=\mathcal{O}\Big(\frac{1}{T}\Big), \tag{20a}$$
$$\mathbf{E}\Big[\frac{1}{n}\sum_{i=1}^{n}\|x_{i,k}-\bar{x}_k\|^2\Big]=\mathcal{O}(p\eta^2), \tag{20b}$$
$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbf{E}[\|\nabla f(\bar{x}_k)\|^2]=\mathcal{O}\Big(\frac{1}{\eta T}\Big). \tag{20c}$$

Moreover, if Assumption 6 also holds, then

$$\mathbf{E}\Big[\frac{1}{n}\sum_{i=1}^{n}\|x_{i,k}-\bar{x}_k\|^2+f(\bar{x}_k)-f^*\Big]=\mathcal{O}(\varepsilon^k),\ \forall k\in\mathbb{N}_+. \tag{21}$$

Remark 5. The result stated in (20c) shows that a stationary point can be found with a rate $\mathcal{O}(p/T)$. This rate is the same as that achieved by the ZO algorithms in Gu et al. (2018), Huang, Gu, Huo, Chen, and Huang (2019), Kozak et al. (2021), Liu, Kailkhura, et al. (2018) and Nesterov and Spokoiny (2017). Although the ZO variance reduced algorithms in Fang et al. (2018) and Ji et al. (2019) and the stochastic direct-search algorithms in Bergou et al. (2020), Bibi et al. (2020) and Gorbunov et al. (2020) achieved a faster rate $\mathcal{O}(1/T)$, these algorithms require three or more samplings at each iteration, while our proposed algorithm requires only two samplings. Moreover, the result stated in (21) shows that a global optimum can be found linearly. The ZO algorithms in Cai et al. (2020), Chen et al. (2020), Ji et al. (2019), Kozak et al. (2021), Nesterov and Spokoiny (2017), Ye et al. (2018) and the stochastic direct-search algorithms in Bergou et al. (2020), Bibi et al. (2020), Golovin et al. (2020), Gorbunov et al. (2020) also achieved linear convergence. However, the algorithms in Bergou et al. (2020), Bibi et al. (2020), Chen et al. (2020), Golovin et al. (2020), Gorbunov et al. (2020), Ji et al. (2019) and Ye et al. (2018) require three or more samplings at each iteration; the P–Ł constant needs to be known in advance in Kozak et al. (2021) and Ji et al. (2019), which is not needed in Theorem 6; and the cost functions in Bergou et al. (2020), Bibi et al. (2020), Cai et al. (2020), Chen et al. (2020), Golovin et al. (2020), Gorbunov et al. (2020), Nesterov and Spokoiny (2017) and Ye et al. (2018) are (restricted) strongly convex, which is stronger than the P–Ł condition used in Theorem 6.

To end this section, we would like to briefly explain the challenges when analyzing the performance of Algorithm 1. Algorithm 1 is simple in the sense that it is a combination of the first-order algorithm proposed in Yi, Zhang, Yang, Chai, and Johansson (2021) with zeroth-order gradient estimators. For such a kind of combination, the standard technique to handle the bias in the ZO gradients is using smoothing function, which is also used in our proofs. However, there still is a gap between the smoothing function and the original function. This gap complicates the proof details, especially under the distributed and stochastic setting. As a result, one needs to make an assumption on the local gradients or the relation between the local and global gradients, such as the Lipschitz assumption, i.e., $\|\nabla f_i(x)\|$ is globally bounded, the bounded variance assumption, i.e., $\|\nabla f_i(x)-\nabla f(x)\|^2$ is globally bounded, or the weaker Assumption 5 used in this paper. Moreover, to the best of our knowledge, how to show linear speedup for distributed ZO algorithms is an open problem in the literature. A key point to show linear speedup is to guarantee that the omitted constants in the dominating term in the convergence rate do not depend on any parameters related to the communication network. In addition, the proofs are much more complicated due to weaker assumptions.

4. Distributed ZO primal algorithm

In this section, we propose a distributed ZO primal algorithm and analyze its convergence rate. Inspired by the distributed first-order (sub)gradient descent algorithm proposed in Nedić and Ozdaglar (2009), we propose the following distributed ZO primal algorithm:

$$x_{i,k+1}=x_{i,k}-\gamma\sum_{j\in\mathcal{N}_i}L_{ij}x_{j,k}-\eta_k g^e_{i,k}, \tag{22}$$

where γ is a positive constant, $\{\eta_k\}$ is a positive sequence, and $g^e_{i,k}$ is the stochastic gradient estimator defined in (5).

We write the distributed random ZO algorithm (22) in pseudo-code as Algorithm 2. Compared with Algorithm 1, in Algorithm 2 each agent only computes the primal variable. Similar results as stated in Theorems 1–6 and Corollary 1 also hold for Algorithm 2. They are given in the online version (Yi, Zhang, Yang, & Johansson, 2021) due to the space limitation.

5. Simulations

In this section, we verify the theoretical results through numerical simulations. Specifically, we evaluate the performance of Algorithms 1 and 2 in generating adversarial examples from black-box deep neural networks (DNNs).

In image classification tasks, DNNs are vulnerable to adversarial examples (Goodfellow, Shlens, & Szegedy, 2015) even under small perturbations, which leads misclassifications. Considering the setting of ZO attacks in Carlini and Wagner (2017) and Liu,

Algorithm 2 Distributed ZO Primal Algorithm

```
Input: positive constant $\gamma$ and positive sequences $\{\eta_k\}$ and $\{\delta_{i,k}\}$.
Initialize: $x_{i,0} \in \mathbb{R}^p, \ \forall i \in [n]$.
for $k = 0, 1, \dots$ do
    for $i = 1, \dots, n$ in parallel do
        Broadcast $x_{i,k}$ to $\mathcal{N}_i$ and receive $x_{j,k}$ from $j \in \mathcal{N}_i$;
        Generate $u_{i,k} \in \mathbb{S}^p$ independently and uniformly at random;
        Generate $\xi_{i,k}$ independently and randomly according to the distribution of $\xi_i$;
        Sample $F_i(x_{i,k}, \xi_{i,k})$ and $F_i(x_{i,k} + \delta_{i,k}u_{i,k}, \xi_{i,k})$;
        Update $x_{i,k+1}$ by (22).
    end for
end for
Output: $\{\boldsymbol{x}_k\}$.
```

Kailkhura, et al. (2018), the model is hidden and no gradient information is available. We treat this task of generating adversarial examples as a ZO optimization problem. The black-box attack loss function (Carlini & Wagner, 2017; Liu, Kailkhura, et al., 2018) is given as

$$f_i(x) = \max\Big\{F_{y_i}\Big(\frac{1}{2}\tanh(\tanh^{-1} 2a_i + x)\Big) - \max_{j \neq y_i}\Big\{F_j\Big(\frac{1}{2}\tanh(\tanh^{-1} 2a_i + x)\Big)\Big\},\ 0\Big\} + c\Big\|\frac{1}{2}\tanh(\tanh^{-1} 2a_i + x) - a_i\Big\|_2^2,$$

where c is a constant, (a_i, y_i) denotes the pair of the ith natural image a_i and its original class label y_i. The output of function $F(z) = \mathrm{col}(F_1(z), \dots, F_m(z))$ is the well-trained model prediction of the input z in all m image classes.

The well-trained DNN model[1] on the MNIST handwritten dataset has 99.4% test accuracy on natural examples (Liu, Kailkhura, et al., 2018). We compare the proposed distributed primal–dual ZO algorithm (Algorithm 1) and distributed primal ZO algorithm (Algorithm 2) with state-of-the-art centralized and distributed ZO algorithms: RSGF (Ghadimi & Lan, 2013), SZO-SPIDER (Fang et al., 2018), ZO-SVRG (Liu, Kailkhura, et al., 2018), SZVR-G (Liu, Cheng, Hsieh, & Tao, 2018), and ZO-SPIDER-Coord (Ji et al., 2019), ZO-GDA (Tang et al., 2020), and ZONE-M (Hajinezhad et al., 2019).

We consider $n = 10$ agents and assume the communication network is generated randomly following the Erdős–Rényi model with probability of 0.4. All the hyper-parameters used in the experiment are given in Yi, Zhang, Yang, and Johansson (2021).

Figs. 1 and 2 show the evolutions of the black-box attack loss achieved by each ZO algorithm with respect to the number of iterations and function value queries, respectively. From these two figures, we can see that our proposed distributed ZO algorithms are as efficient as ZO-GDA (Tang et al., 2020) in terms of both convergence rate and sampling complexity, and more efficient than the other algorithms. The least ℓ_2 distortions of the successful adversarial perturbations are listed in Table 1. We can see that the adversarial examples generated by the distributed algorithms in general have slightly larger ℓ_2 distortions than those generated by the centralized algorithms. A comparison of generated adversarial examples from the DNN on the MNIST dataset is summarized in a table provided in Yi, Zhang, Yang, and Johansson (2021).

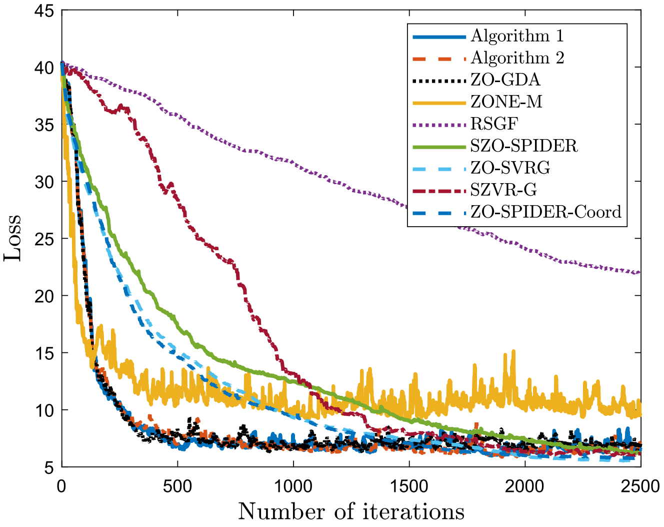


Fig. 1. Evolutions of the black-box attack loss with respect to the number of iterations.

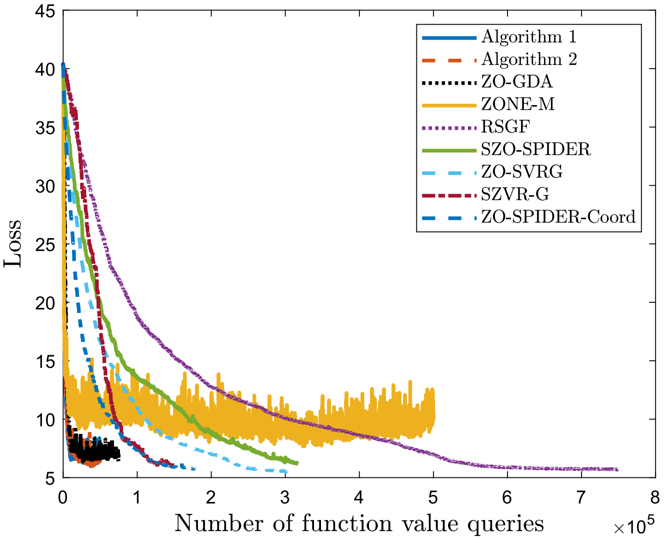


Fig. 2. Evolutions of the black-box attack loss with respect to the number of function value queries.

Table 1
Distortion.

Algorithm	Distributed	ℓ_2 distortion
Algorithm 1	✔	6.44
Algorithm 2	✔	5.77
ZO-GDA	✔	7.23
ZONE-M	✔	9.96
RSGF	✗	5.69
SZO-SPIDER	✗	6.19
ZO-SVRG	✗	4.76
SZVR-G	✗	5.16
ZO-SPIDER-Coord	✗	5.76

In order to verify the result that linear speedup convergence is achieved with respect to the number of agents, we also consider $n = 100$ agents. To illustrate the linear speedup results in a more clear manner, we plot the loss in log scale and draw the extensive lines along the convergence lines in Fig. 3. The slopes of the 10-node lines (blue and red lines) are approximately −0.025 and the slopes of the 100-node lines (blue and red dash lines) are approximately −0.079, which suggests linear speedup

[1] https://github.com/carlini/nn_robust_attacks

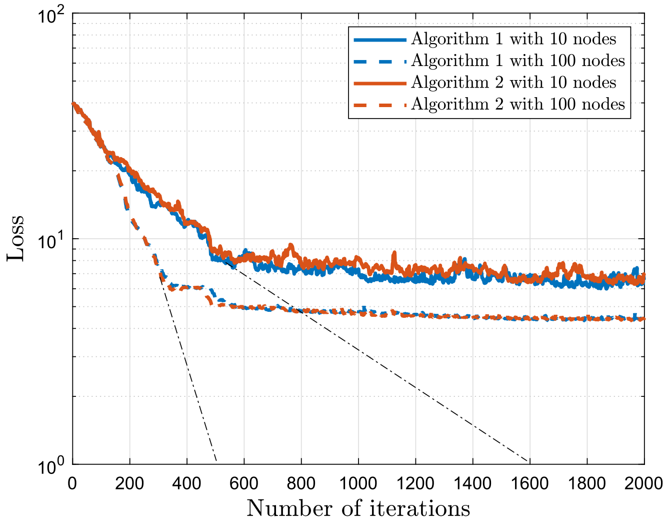


Fig. 3. Evolutions of the black-box attack loss with respect to the number of iterations when using different numbers of agents. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

since $-0.079/\sqrt{10} \approx -0.025$. This simulation shows that linear speedup is achieved by our proposed algorithms even though the optimization problem is nonsmooth.

6. Conclusions

In this paper, we studied stochastic distributed nonconvex optimization with ZO information feedback. We proposed two distributed ZO algorithms and analyzed their convergence properties. More specifically, linear speedup convergence rate $\mathcal{O}(\sqrt{p/(nT)})$ was established for smooth nonconvex cost functions under arbitrarily connected communication networks. The convergence rate was improved to $\mathcal{O}(p/(nT))$ when the global cost function satisfies the P–Ł condition. It was also shown that the output of the proposed algorithms linearly converges to a neighborhood of a global optimum. Interesting directions for future work include establishing faster convergence with reduced sampling complexity by using variance reduction techniques, and considering communication reduction with asynchronous, periodic, or compressed communication.

References

Audet, C., & Hare, W. (2017). *Derivative-free and blackbox optimization*. Springer.

Bach, F., & Perchet, V. (2016). Highly-smooth zero-th order online optimization. In *Conference on learning theory* (pp. 257–283).

Balasubramanian, K., & Ghadimi, S. (2018). Zeroth-order (non)-convex stochastic optimization via conditional gradient and gradient updates. In *Advances in neural information processing systems* (pp. 3455–3464).

Bergou, E. H., Gorbunov, E., & Richtárik, P. (2020). Stochastic three points method for unconstrained smooth minimization. *SIAM Journal on Optimization*, *30*, 2726–2749.

Beznosikov, A., Gorbunov, E., & Gasnikov, A. (2020). Derivative-free method for composite optimization with applications to decentralized distributed optimization. *IFAC-PapersOnLine*, *53*, 4038–4043.

Bibi, A., Bergou, E. H., Sener, O., Ghanem, B., & Richtárik, P. (2020). A stochastic derivative-free optimization method with importance sampling: Theory and learning to control. In *International conference on learning representations*.

Cai, H., Mckenzie, D., Yin, W., & Zhang, Z. (2020). Zeroth-order regularized optimization (ZORO): Approximately sparse gradients and adaptive sampling. arXiv:2003.13001.

Carlini, N., & Wagner, D. (2017). Towards evaluating the robustness of neural networks. In *IEEE symposium on security and privacy* (pp. 39–57).

Chen, X., Liu, S., Xu, K., Li, X., Lin, X., Hong, M., et al. (2019). ZO-AdaMM: Zeroth-order adaptive momentum method for black-box optimization. In *Advances in neural information processing systems* (pp. 7204–7215).

Chen, Y., Orvieto, A., & Lucchi, A. (2020). An accelerated DFO algorithm for finite-sum convex functions. In *International conference on machine learning* (pp. 1681–1690).

Chen, P. Y., Zhang, H., Sharma, Y., Yi, J., & Hsieh, C. J. (2017). ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *ACM workshop on artificial intelligence and security* (pp. 15–26).

Conn, A. R., Scheinberg, K., & Vicente, L. N. (2009a). Global convergence of general derivative-free trust-region algorithms to first- and second-order critical points. *SIAM Journal on Optimization*, *20*, 387–415.

Conn, A. R., Scheinberg, K., & Vicente, L. N. (2009b). Introduction to derivative-free optimization. In *MPS-SIAM series on optimization*. SIAM Philadelphia.

Duchi, J. C., Jordan, M. I., Wainwright, M. J., & Wibisono, A. (2015). Optimal rates for zero-order convex optimization: The power of two function evaluations. *IEEE Transactions on Information Theory*, *61*, 2788–2806.

Fang, C., Li, C. J., Lin, Z., & Zhang, T. (2018). Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. In *Advances in neural information processing systems* (pp. 689–699).

Fazel, M., Ge, R., Kakade, S., & Mesbahi, M. (2018). Global convergence of policy gradient methods for the linear quadratic regulator. In *International conference on machine learning* (pp. 1467–1476).

Gao, H., & Huang, H. (2020). Can stochastic zeroth-order Frank–Wolfe method converge faster for non-convex problems? In *International conference on machine learning* (pp. 3377–3386).

Gao, X., Jiang, B., & Zhang, S. (2018). On the information-adaptive variants of the ADMM: An iteration complexity perspective. *Journal of Scientific Computing*, *76*, 327–363.

Ghadimi, S., & Lan, G. (2013). Stochastic first- and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, *23*, 2341–2368.

Ghadimi, S., Lan, G., & Zhang, H. (2016). Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization. *Mathematical Programming*, *155*, 267–305.

Golovin, D., Karro, J., Kochanski, G., Lee, C., Song, X., et al. (2020). Gradientless descent: High-dimensional zeroth-order optimization. In *International conference on learning representations*.

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. In *International conference on learning representations*.

Gorbunov, E., Bibi, A., Sener, O., Bergou, E. H., & Richtárik, P. (2020). A stochastic derivative free optimization method with momentum. In *International conference on learning representations*.

Gorbunov, E., Dvurechensky, P., & Gasnikov, A. (2018). An accelerated method for derivative-free smooth stochastic convex optimization. arXiv:1802.09022.

Gratton, C., Venkategowda, N. K., Arablouei, R., & Werner, S. (2021). Privacy-preserved distributed learning with zeroth-order optimization. *IEEE Transactions on Information Forensics and Security*, *17*, 265–279.

Gu, B., Huo, Z., Deng, C., & Huang, H. (2018). Faster derivative-free stochastic algorithm for shared memory machines. In *International conference on machine learning* (pp. 1812–1821).

Hajinezhad, D., Hong, M., & Garcia, A. (2019). ZONE: ZEroth-order nonconvex multiagent optimization over networks. *IEEE Transactions on Automatic Control*, *64*, 3995–4010.

Hajinezhad, D., & Zavlanos, M. M. (2018). Gradient-free multi-agent nonconvex nonsmooth optimization. In *IEEE conference on decision and control* (pp. 4939–4944).

Hooke, R., & Jeeves, T. A. (1961). Direct search solution of numerical and statistical problems. *Journal of the ACM*, *8*, 212–229.

Huang, F., Gao, S., Chen, S., & Huang, H. (2019). Zeroth-order stochastic alternating direction method of multipliers for nonconvex nonsmooth optimization. In *International conference on artificial intelligence and statistics* (pp. 2549–2555).

Huang, F., Gao, S., Pei, J., & Huang, H. (2019). Nonconvex zeroth-order stochastic ADMM methods with lower function query complexity. arXiv:1907.13463.

Huang, F., Gu, B., Huo, Z., Chen, S., & Huang, H. (2019). Faster gradient-free proximal stochastic methods for nonconvex nonsmooth optimization. In *AAAI conference on artificial intelligence* (pp. 1503–1510).

Huang, F., Tao, L., & Chen, S. (2020). Accelerated stochastic gradient-free and projection-free methods. In *International conference on machine learning* (pp. 4519–4530).

Jakovetić, D., Bajović, D., Xavier, J., & Moura, J. M. (2020). Primal–dual methods for large-scale and distributed convex optimization and data analytics. *Proceedings of the IEEE*, *108*, 1923–1938.

Ji, K., Wang, Z., Zhou, Y., & Liang, Y. (2019). Improved zeroth-order variance reduced algorithms and analysis for nonconvex optimization. In *International conference on machine learning* (pp. 3100–3109).

Jin, C., Liu, L. T., Ge, R., & Jordan, M. I. (2018). On the local minima of the empirical risk. In *Advances in neural information processing systems* (pp. 4896–4905).

Karimi, H., Nutini, J., & Schmidt, M. (2016). Linear convergence of gradient and proximal-gradient methods under the Polyak–Łojasiewicz condition. In *Joint European conference on machine learning and knowledge discovery in databases* (pp. 795–811).

Kazemi, E., & Wang, L. (2018). A proximal zeroth-order algorithm for nonconvex nonsmooth problems. In *Annual Allerton conference on communication, control, and computing* (pp. 64–71).

Koloskova, A., Stich, S., & Jaggi, M. (2019). Decentralized stochastic optimization and gossip algorithms with compressed communication. In *International conference on machine learning* (pp. 3478–3487).

Kozak, D., Becker, S., Doostan, A., & Tenorio, L. (2021). A stochastic subspace approach to gradient-free optimization in high dimensions. *Computational Optimization and Applications*, *79*, 339–368.

Larson, J., Menickelly, M., & Wild, S. M. (2019). Derivative-free optimization methods. *Acta Numerica*, *28*, 287–404.

Li, Z., & Li, J. (2018). A simple proximal stochastic gradient method for nonsmooth nonconvex optimization. In *Advances in neural information processing systems* (pp. 5569–5579).

Lian, X., Zhang, H., Hsieh, C. J., Huang, Y., & Liu, J. (2016). A comprehensive linear speedup analysis for asynchronous stochastic parallel optimization from zeroth-order to first-order. In *Advances in neural information processing systems* (pp. 3054–3062).

Lian, X., Zhang, C., Zhang, H., Hsieh, C. J., Zhang, W., & Liu, J. (2017). Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent. In *Advances in neural information processing systems* (pp. 5330–5340).

Liu, S., Chen, P. Y., Chen, X., & Hong, M. (2019). signSGD via zeroth-order oracle. In *International conference on learning representations*.

Liu, L., Cheng, M., Hsieh, C. J., & Tao, D. (2018). Stochastic zeroth-order optimization via variance reduction method. arXiv:1805.11811.

Liu, S., Kailkhura, B., Chen, P. Y., Ting, P., Chang, S., & Amini, L. (2018). Zeroth-order stochastic variance reduction for nonconvex optimization. In *Advances in neural information processing systems* (pp. 3727–3737).

Liu, S., Li, X., Chen, P. Y., Haupt, J., & Amini, L. (2018). Zeroth-order stochastic projected gradient descent for nonconvex optimization. In *IEEE global conference on signal and information processing* (pp. 1179–1183).

Marazzi, M., & Nocedal, J. (2002). Wedge trust region methods for derivative free optimization. *Mathematical Programming*, *91*, 289–305.

Matyas, J. (1965). Random optimization. *Automation and Remote Control*, *26*, 246–253.

Nazari, P., Tarzanagh, D. A., & Michailidis, G. (2020). Adaptive first- and zeroth-order methods for weakly convex stochastic optimization problems. arXiv:2005.09261.

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

Nedić, A., Olshevsky, A., Shi, W., & Uribe, C. A. (2017). Geometrically convergent distributed optimization with uncoordinated step-sizes. In *American control conference* (pp. 3950–3955).

Nedić, A., & Ozdaglar, A. (2009). Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, *54*, 48–61.

Nelder, J. A., & Mead, R. (1965). A simplex method for function minimization. *The Computer Journal*, *7*, 308–313.

Nesterov, Y., & Spokoiny, V. (2017). Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, *17*, 527–566.

Pang, Y., & Hu, G. (2020). Randomized gradient-free distributed optimization methods for a multi-agent system with unknown cost function. *IEEE Transactions on Automatic Control*, *65*, 333–340.

Qu, G., & Li, N. (2018). Harnessing smoothness to accelerate distributed optimization. *IEEE Transactions on Control of Network Systems*, *5*, 1245–1260.

Qu, G., & Li, N. (2020). Accelerated distributed nesterov gradient descent. *IEEE Transactions on Automatic Control*, *65*, 2566–2581.

Sahu, A. K., Jakovetić, D., Bajović, D., & Kar, S. (2018a). Communication-efficient distributed strongly convex stochastic optimization: Non-asymptotic rates. arXiv preprint arXiv:1809.02920.

Sahu, A. K., Jakovetić, D., Bajović, D., & Kar, S. (2018b). Distributed zeroth order optimization over random networks: A Kiefer–Wolfowitz stochastic approximation approach. In *IEEE conference on decision and control* (pp. 4951–4958).

Sahu, A. K., & Kar, S. (2020). Decentralized zeroth-order constrained stochastic optimization algorithms: Frank–Wolfe and variants with applications to black-box adversarial attacks. *Proceedings of the IEEE*, *108*, 1890–1905.

Sahu, A. K., Zaheer, M., & Kar, S. (2019). Towards gradient free and projection free stochastic optimization. In *International conference on artificial intelligence and statistics* (pp. 3468–3477).

Scheinberg, K., & Toint, P. L. (2010). Self-correcting geometry in model-based algorithms for derivative-free unconstrained optimization. *SIAM Journal on Optimization*, *20*, 3512–3532.

Shamir, O. (2013). On the complexity of bandit and derivative-free stochastic convex optimization. In *Conference on learning theory* (pp. 3–24).

Shamir, O. (2017). An optimal algorithm for bandit and zero-order convex optimization with two-point feedback. *Journal of Machine Learning Research*, *18*, 1–11.

Shi, W., Ling, Q., Wu, G., & Yin, W. (2015). EXTRA: AN exact first-order algorithm for decentralized consensus optimization. *SIAM Journal on Optimization*, *25*, 944–966.

Tang, Y., Zhang, J., & Li, N. (2020). Distributed zero-order algorithms for nonconvex multiagent optimization. *IEEE Transactions on Control of Network Systems*, *8*, 269–281.

Vlatakis-Gkaragkounis, E. V., Flokas, L., & Piliouras, G. (2019). Efficiently avoiding saddle points with zero order methods: No gradients required. In *Advances in neural information processing systems* (pp. 10066–10077).

Wang, Y., Du, S., Balakrishnan, S., & Singh, A. (2018). Stochastic zeroth-order optimization in high dimensions. In *International conference on artificial intelligence and statistics* (pp. 1356–1365).

Wang, Y., Zhao, W., Hong, Y., & Zamani, M. (2019). Distributed subgradient-free stochastic optimization algorithm for nonsmooth convex functions over time-varying networks. *SIAM Journal on Control and Optimization*, *57*, 2821–2842.

Yang, T., Yi, X., Wu, J., Yuan, Y., Wu, D., Meng, Z., et al. (2019). A survey of distributed optimization. *Annual Reviews in Control*, *47*, 278–305.

Ye, H., Huang, Z., Fang, C., Li, C. J., & Zhang, T. (2018). Hessian-aware zeroth-order optimization for black-box adversarial attack. arXiv:1812.11377.

Yi, X., Zhang, S., Yang, T., Chai, T., & Johansson, K. H. (2021). Linear convergence of first- and zeroth-order primal–dual algorithms for distributed nonconvex optimization. *IEEE Transactions on Automatic Control*, in press.

Yi, X., Zhang, S., Yang, T., & Johansson, K. H. (2021). Zeroth-order algorithms for stochastic distributed nonconvex optimization. arXiv:2106.02958v3.

Yu, Z., Ho, D. W., & Yuan, D. (2022). Distributed randomized gradient-free mirror descent algorithm for constrained optimization. *IEEE Transactions on Automatic Control*, *67*, 957–964.

Yu, H., Jin, R., & Yang, S. (2019). On the linear speedup analysis of communication efficient momentum SGD for distributed non-convex optimization. In *International conference on machine learning* (pp. 7184–7193).

Yuan, D., & Ho, D. W. (2014). Randomized gradient-free method for multiagent optimization over time-varying networks. *IEEE Transactions on Neural Networks and Learning Systems*, *26*, 1342–1347.

Yuan, D., Xu, S., & Lu, J. (2015). Gradient-free method for distributed multiagent optimization via push-sum algorithms. *International Journal of Robust and Nonlinear Control*, *25*, 1569–1580.

Zhang, H., & Cheng, L. (2015). Restricted strong convexity and its applications to convergence analysis of gradient-type methods in convex optimization. *Optimization Letters*, *9*, 961–979.

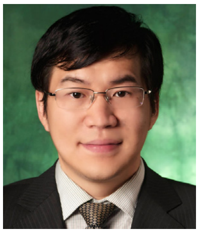
Zhang, Y., Zhou, Y., Ji, K., & Zavlanos, M. M. (2022). A new one-point residual-feedback oracle for black-box learning and control. *Automatica*, *136*, Article 110006.

Xinlei Yi received the Ph.D. degree in electrical engineering from the School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology in 2020 and now is a postdoc at the same university. He received B.S. and M.S. degrees in mathematics from China University of Geoscience and Fudan University, in 2011 and 2014, respectively.

His current research interests include online optimization, distributed optimization, and event-triggered control.

Shengjun Zhang received the B.Eng. degree in Automation of Honors Program from China Agricultural University, Beijing, China, in 2014, and the M.S. degree in Electrical Engineering from New York University, New York, New York, USA, in 2017. He is currently working toward a Ph.D. degree with the Department of Electrical Engineering in the College of Engineering, University of North Texas, Denton, Texas, USA.

His current research interests include distributed optimization, statistical learning, and Sparse PCA.



Tao Yang is a Professor at the State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University. He was an Assistant Professor at the Department of Electrical Engineering, University of North Texas, Denton, USA, from 2016–2019. He received the Ph.D. degree in electrical engineering from Washington State University in 2012. Between August 2012 and August 2014, he was an ACCESS postdoctoral researcher with the ACCESS Linnaeus Centre, Royal Institute of Technology, Sweden. He then joined the Pacific Northwest National Laboratory as a postdoc, and was promoted to Scientist/Engineer II in 2015.

His research interests include industrial artificial intelligence, integrated optimization and control, distributed control and optimization with applications to process industries, cyber physical systems, networked control systems, and multi-agent systems. He is an Associate Editor for IEEE Transactions on Control Systems Technology, IEEE Transactions on Neural Networks and Learning Systems, and IEEE/CAA Journal of Automatica Sinica. He currently is a member of the Technical Committee on Nonlinear Systems and Control, the Technical Committee on Networks and Communication Systems, and the Technical Committee on Smart Grids of the IEEE Control Systems Society, a member of the IEEE Control Systems Society Conference Editorial Board, and a member of the IFAC Technical Committee on Large Scale Complex Systems. He received Ralph E. Powe Junior Faculty Enhancement Award and Best Student Paper award (as an advisor) of the 14th IEEE International Conference on Control & Automation in 2018.

Karl H. Johansson is Professor at the School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology. He received M.Sc. and Ph.D. degrees from Lund University. He has held visiting positions at UC Berkeley, Caltech, NTU, HKUST Institute of Advanced Studies, and NTNU.

His research interests are in networked control systems, cyber–physical systems, and applications in transportation, energy, and automation. He has served on the IEEE Control Systems Society Board of Governors, the IFAC Executive Board, and the European Control Association Council. He has received several best paper awards and other distinctions from IEEE and ACM. He has been awarded Distinguished Professor with the Swedish Research Council and Wallenberg Scholar with the Knut and Alice Wallenberg Foundation. He has received the Future Research Leader Award from the Swedish Foundation for Strategic Research and the triennial Young Author Prize from IFAC. He is Fellow of the IEEE and the Royal Swedish Academy of Engineering Sciences, and he is IEEE Control Systems Society Distinguished Lecturer.