Distributed Zero-Order Algorithms for Nonconvex Multiagent Optimization

Yujie Tang, *Member, IEEE*, Junshan Zhang, *Fellow, IEEE*, and Na Li, *Member, IEEE*

***Abstract*—Distributed multiagent optimization finds many applications in distributed learning, control, estimation, etc. Most existing algorithms assume knowledge of first-order information of the objective and have been analyzed for convex problems. However, there are situations where the objective is nonconvex, and one can only evaluate the function values at finitely many points. In this article, we consider derivative-free distributed algorithms for nonconvex multiagent optimization, based on recent progress in zero-order optimization. We develop two algorithms for different settings, provide detailed analysis of their convergence behavior, and compare them with existing centralized zero-order algorithms and gradient-based distributed algorithms.**

***Index Terms*—Distributed optimization, nonconvex optimization, zero-order information.**

I. Introduction

CONSIDER a set of n agents connected over a network, each of which is associated with a smooth local objective function f_i that can be nonconvex. The goal is to solve the optimization problem

$$\min_{x\in\mathbb{R}^d} \quad f(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x)$$

with the restriction that f_i is only known to agent i and each agent can exchange information only with its neighbors in the network during the optimization procedure. We focus on the situation where only zero-order information of f_i is available to agent i.

Distributed multiagent optimization lies at the core of a wide range of applications, and a large body of literature has been contributed toward distributed multiagent optimization algorithms. One line of research combines (sub)gradient-based methods with a consensus/averaging scheme, where each iteration of a local agent consists of one or multiple consensus steps and a local gradient evaluation step. It has been shown that for convex functions, the convergence rates of distributed gradient-based algorithms can match or nearly match those of centralized gradient-based algorithms. Specifically, papers [3] and [10] proposed and analyzed consensus-based decentralized gradient descent (DGD) algorithms with $O(\log t/\sqrt{t})$ convergence for nonsmooth convex functions; papers [8], [11], and [12] employed the *gradient tracking* scheme and showed that the DGD with gradient tracking achieves $O(1/t)$ convergence for smooth convex functions and linear convergence for strongly convex functions; and paper [13] employed Nesterov's gradient descent method and showed $O(1/t^{1.4-\epsilon})$ convergence for smooth convex functions and improved linear convergence for strongly convex functions where ϵ is an arbitrarily small positive number. Besides convergence rates, some works have additional focuses, such as time-varying/directed graphs [14], uncoordinated step sizes [15], and stochastic (sub)gradient [16].

While distributed convex optimization has broad applicability, nonconvex problems also appear in important applications, such as distributed learning [17], robotic networks [18], and operation of wind farms [19]. Several works have considered nonconvex multiagent optimization and developed various distributed gradient-based methods to converge to stationary points with convergence rate analysis, e.g., [4], [6], [7], [20]. We notice that for smooth functions, either convex or nonconvex, in general DGD with *gradient tracking* converges faster than the method without gradient tracking, and its convergence rate has the same big-O dependence on the number of iterations as the centralized vanilla gradient descent method (see Table I).

Further, there has been an increasing interest in zero-order optimization, where one does not have access to the gradient of the objective. Such situations can occur, for example, when only black-box procedures are available for computing the values of the functional characteristics of the problem, or when resource limitations restrict the use of fast or automatic differentiation techniques. Many existing works [9], [21]–[24] on zero-order optimization are based on constructing gradient estimators using finitely many function evaluations, e.g., gradient estimator based on Kiefer–Wolfowitz scheme [21] by using $2d$-point function evaluations where d is the dimension of the problem. However, this estimator does not scale up well with high-dimensional problems. Paper [22] proposed and analyzed a single-point gradient estimator, and paper [23] further studied the convergence rate for highly smooth objectives. Paper [9] proposed two-point gradient estimators and showed that the

Manuscript received January 6, 2020; revised January 7, 2020, May 31, 2020, and August 22, 2020; accepted August 22, 2020. Date of publication September 16, 2020; date of current version February 26, 2021. The work was supported by the NSF CAREER: ECCS-1553407, AFOSR YIP: FA9550-18-1-0150, and ONR YIP: N00014-19-1-2217 programs. This article was presented in part at the 57th Annual Allerton Conference on Communication, Control, and Computing, Monticello, IL, USA, September 2019. Recommended by Associate Editor A. Olshevsky. *(Corresponding author: Yujie Tang.)*

Yujie Tang and Na Li are with the School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138 USA (e-mail: yujietang@seas.harvard.edu; nali@seas.harvard.edu).

Junshan Zhang is with the School of Electrical, Computer, and Energy Engineering, Arizona State University, Tempe, AZ 85287 USA (e-mail: junshan.zhang@asu.edu).

Digital Object Identifier 10.1109/TCNS.2020.3024321

TABLE I
COMPARISON OF DIFFERENT ALGORITHMS FOR DISTRIBUTED OPTIMIZATION AND ZERO-ORDER OPTIMIZATION

		smooth	gradient dominated
distributed zero-order (nonconvex)	Alg. 1, this paper (2-point + DGD)	$O\left(\sqrt{\frac{d}{m}}\log m\right)$	$O\left(\frac{d}{m}\right)$
	Alg. 2, this paper ($2d$-point + gradient tracking)	$O\left(\frac{d}{m}\right)$	$O\left(\left[1-c(1-\rho^2)^2\left(\frac{\mu}{L}\right)^{\frac{4}{3}}\right]^{m/d}\right)$
	ZONE [2]	$O\left(\frac{\gamma(d)}{\sqrt{M}}\right)$	—
distributed first-order	DGD	$O\left(\frac{\log t}{\sqrt{t}}\right)$ [3], [4] (convex) $O\left(\frac{1}{\sqrt{T}}\right)$ [6] (nonconvex)	$O\left(\frac{1}{t}\right)$ [5] (strongly convex)
	gradient tracking	$O\left(\frac{1}{t}\right)$ [7] (nonconvex)	$O\left(\left[1-c(1-\rho)^2\left(\frac{\mu}{L}\right)^{\frac{3}{2}}\right]^t\right)$ [8] (strongly convex)
centralized zero-order	[9] (2-point estimator)	$O\left(\frac{d}{m}\right)$ (nonconvex)	$O\left(\left[1-\frac{c}{d}\frac{\mu}{L}\right]^m\right)$ (strongly convex)

Note: The table summarizes best known convergence rates for deterministic nonconvex unconstrained optimization with 1) smooth, 2) gradient dominated objectives. The convex counterparts are listed if results for nonconvex cases have not been established.
m denotes the number of function value queries, t denotes the number of iterations, d denotes the dimension of the decision variable, c's represent numerical constants that can be different for different algorithms.
M denotes the total number of function value queries and T denotes the total number of iterations provided before the optimization procedure. The rates in [2] and [6] assume constant step sizes chosen based on M or T.
The listed convergence rates are the ergodic rates of $\|\nabla f\|^2$ for the smooth case, and the objective error rates for the gradient dominated case, respectively.
The rates provided in [2] do not include explicit dependence on (d); we use $\gamma(d)$ to denote this dependence.
The cited results in this table may apply to more general settings (e.g., stochastic gradients [5], [6]).
We do not include algorithms with Nesterov-type acceleration in this comparison.

convergence rates of the resulting algorithms are comparable to their first-order counterparts (see Table I). For instance, gradient descent with two-point gradient estimators converges with a rate of $O(d/m)$, where m denotes the number of function value queries. Papers [24] and [25] showed that two-point gradient estimators achieve the optimal rate $O(\sqrt{d/m})$ of stochastic zero-order convex optimization.

Some recent works have started to combine zero-order and distributed optimization methods [2], [26], [27]. For example, paper [2] proposed the ZONE algorithm for stochastic nonconvex problems based on the method of multipliers. Paper [26] proposed a distributed zero-order algorithm over random networks and established its convergence for strongly convex objectives. Paper [27] considered distributed zero-order methods for constrained convex optimization. However, there are still many questions remaining to be studied in distributed zero-order optimization. In particular, *how do zero-order and distributed methods affect the performance of each other*, and *could their fundamental structural properties be kept when combining the two?* For instance, it would be ideal if we could combine both 2-point zero-order methods with DGD with gradient tracking and maintain the nice properties for both methods, leading to an "optimal" distributed zero-order algorithm if possible. This is unclear *a priori*, and indeed, as we shall show later, the 2-point gradient estimator and DGD with gradient tracking do not reconcile with each other well.

Contributions. Motivated by the above observations, we propose two distributed zero-order algorithms: Algorithm 1 is based on the 2-point estimator and DGD; and Algorithm 2 is based on the $2d$-point gradient estimator and DGD with gradient tracking. We analyze the performance of the two algorithms for deterministic nonconvex optimization, and compare their convergence rates with their distributed first-order and centralized zero-order counterparts. The convergence rates of the two algorithms are summarized in Table I. Specifically, it can be seen that the rates of Algorithm 1 are comparable with the first-order DGD but are inferior to the centralized zero-order method; the rates of Algorithm 2 are comparable with the centralized zero-order method and the first-order DGD with gradient tracking. On the other hand, Algorithm 1 uses the 2-point gradient estimator that requires only two function value queries, whereas Algorithm 2 employs the $2d$-point gradient estimator whose computation involves $2d$ function value queries, indicating that Algorithm 1 could be favored for high-dimensional problems even though its convergence is slower asymptotically, whereas Algorithm 2 could handle problems of relatively low dimensions better with faster convergence. These results shed light on how zero-order evaluations affect distributed optimization and how the presence of network structure affects zero-order algorithms. Different problems and different computation requirements would favor different integration of zero-order methods and distributed methods.

Compared to the existing literature on distributed zero-order optimization, our Algorithm 1 is similar to the algorithms proposed in [26] and [27], but our analysis assumes nonconvex objectives and considers gradient dominated functions. While

paper [2] analyzed the performance of the ZONE algorithm for unconstrained nonconvex problems, we shall see that our Algorithm 1 achieves comparable convergence behavior with ZONE-M, and Algorithm 2 converges faster than ZONE-M in the deterministic setting due to the use of the gradient tracking technique. A more detailed comparison will be given in Section III-D.

Notation: We denote the ℓ_2-norm of vectors and matrices by $\|\cdot\|$. The standard basis of $\mathbb{R}^d$ will be denoted by $\{e_k\}_{k=1}^d$. We let $\mathbf{1}_n \in \mathbb{R}^n$ denote the vector of all ones. We let $\mathbb{B}_d$ denote the closed unit ball in $\mathbb{R}^d$, and let $\mathbb{S}_{d-1} := \{x \in \mathbb{R}^d : \|x\| = 1\}$ denote the unit sphere. The uniform distributions over $\mathbb{B}_d$ and $\mathbb{S}_{d-1}$ will be denoted by $\mathcal{U}(\mathbb{B}_d)$ and $\mathcal{U}(\mathbb{S}_{d-1})$. I_d denotes the $d \times d$ identity matrix. For two matrices $A = [a_{ij}] \in \mathbb{R}^{p\times q}$ and $B = [b_{ij}] \in \mathbb{R}^{r\times s}$, their tensor product $A \otimes B$ is

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1q}B \\ \vdots & \ddots & \vdots \\ a_{p1}B & \cdots & a_{pq}B \end{bmatrix} \in \mathbb{R}^{pr\times qs}.$$

II. Formulation and Algorithms

A. Problem Formulation

Let $\mathcal{N} = \{1, 2, \ldots, n\}$ be the set of agents. Suppose the agents are connected by a communication network, whose topology is represented by an undirected, connected graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ where the edges in $\mathcal{E}$ represent communication links.

Each agent i is associated with a local objective function $f_i : \mathbb{R}^d \to \mathbb{R}$. The goal of the agents is to collaboratively solve the optimization problem

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

We assume that at each time step, agent i can only query the function values of f_i at finitely many points, and can only communicate with its neighbors. Similar to [9] and other works on zero-order optimization, we assume a deterministic setting where the queries of the function values are *noise-free* and *error-free*. The analysis of the deterministic setting will provide a baseline for extension to stochastic optimization, which we leave as future work.

The following definitions will be useful later in this article.

Definition 1:

1) A function $f : \mathbb{R}^d \to \mathbb{R}$ is said to be L-*smooth* if f is continuously differentiable and satisfies
$$\|\nabla f(x) - \nabla f(y)\| \le L\|x - y\| \qquad \forall x, y \in \mathbb{R}^d.$$
2) A function $f : \mathbb{R}^d \to \mathbb{R}$ is said to be G-*Lipschitz* if
$$|f(x) - f(y)| \le G\|x - y\| \qquad \forall x, y \in \mathbb{R}^d.$$
3) A function $f : \mathbb{R}^d \to \mathbb{R}$ is said to be μ-*gradient dominated* if f is differentiable, has a global minimizer x^*, and
$$2\mu(f(x) - f(x^*)) \le \|\nabla f(x)\|^2 \qquad \forall x \in \mathbb{R}^d.$$

The notion of gradient domination is also known as Polyak–Łojasiewicz inequality, first introduced by papers [28] and [29]. It can be viewed as a nonconvex analogy of strong convexity, as the centralized vanilla gradient descent achieves linear convergence for gradient dominated objective functions. The gradient domination condition has been frequently discussed in nonconvex optimization [28], [30]. Also, nonconvex but gradient dominated objective functions appear in many applications, e.g., linear quadratic control problems [31] and deep linear neural networks [32].

B. Preliminaries on Zero-Order and Distributed Optimization

We present some preliminaries to motivate our algorithm development.

Zero-order optimization based on gradient estimation: In zero-order optimization, one tries to minimize a function with the limitation that only function values at finitely many points may be obtained. One basic approach of designing zero-order optimization algorithms is to construct gradient estimators from zero-order information and substitute them for the true gradients. Here, we introduce two types of zero-order gradient estimators for the noiseless setting.

1) The $2d$-*point gradient estimator* is given by
$$\mathsf{G}_f^{(2d)}(x; u) = \sum_{k=1}^{d} \frac{f(x + ue_k) - f(x - ue_k)}{2\,u} e_k \tag{2}$$
where u is some given positive number. Basically, it approximates the gradient $\nabla f(x)$ by taking finite differences along d orthogonal directions, and can be viewed as a noise-free version of the classical Kiefer–Wolfowitz type method [21]. Given an L-smooth function $f : \mathbb{R}^d \to \mathbb{R}$, it can be shown that
$$\|\mathsf{G}_f^{(2d)}(x; u) - \nabla f(x)\| \le \frac{1}{2}uL\sqrt{d}$$
for any $x \in \mathbb{R}^d$. The right-hand side decreases to zero as $u \to 0$. In other words, $\mathsf{G}_f^{(2d)}(x; u)$ can be arbitrarily close to $\nabla f(x)$ (as long as the finite differences can be evaluated accurately). One drawback of this estimator is that it requires $2d$ zero-order queries, which may not be computationally efficient for high-dimensional problems.
2) The *2-point gradient estimator* is given by
$$\mathsf{G}_f^{(2)}(x; u, z) := d \cdot \frac{f(x + uz) - f(x - uz)}{2u} z \tag{3}$$
where $z \in \mathbb{R}^d$ is a random vector that is sampled from the distribution $\mathcal{U}(\mathbb{S}_{d-1})$, and $u > 0$ is a given positive number. The following proposition indicates that when z is uniformly sampled from the sphere $\mathbb{S}_{d-1}$, the expectation of $\mathsf{G}_f^{(2)}(x; u, z)$ is the gradient of a "locally averaged" version of f.

Proposition 1 (see [22]): Suppose $f : \mathbb{R}^d \to \mathbb{R}$ is L-smooth. Then, for any $u > 0$ and $x \in \mathbb{R}^d$, we have
$$\mathbb{E}_{z\sim\mathcal{U}(\mathbb{S}_{d-1})}\left[\mathsf{G}_f^{(2)}(x; u, z)\right] = \nabla f^u(x)$$
where $f^u(x) := \mathbb{E}_{y\sim\mathcal{U}(\mathbb{B}_d)}[f(x + uy)]$.

It has been shown in [9] that if we substitute $\mathsf{G}_f^{(2)}(x;u,z)$ for the gradient in the gradient descent algorithm, we have

$$\frac{1}{t}\sum_{\tau=0}^{t-1}\|\nabla f(x_\tau)\|^2 = O\left(\frac{d}{m}\right)$$

for nonconvex smooth objectives, and

$$f(x_t) - f^* = O\left(\left[1 - c\,\frac{\mu/L}{d}\right]^m\right)$$

for smooth and strongly convex objectives, where x_τ denotes the τth iterate and m denotes the number of zero-order queries in t iterations (see Table I). These rates are comparable to the rates of the (centralized) vanilla gradient descent method, i.e., $O(1/t)$ for nonconvex smooth objectives and linear convergence for smooth and strongly convex objectives.

Distributed optimization: In this article, we mainly focus on consensus-based algorithms for distributed optimization, where each agent maintains a local copy of the global variables, and weighs its neighbors' information to update its own local variable. Specifically, for a time-invariant and bidirectional communication network, we introduce a consensus matrix $W = [W_{ij}] \in \mathbb{R}^{n\times n}$ that satisfies the following assumption.

Assumption 1:

1) W is a doubly stochastic matrix.
2) $W_{ii} > 0$ for all $i \in \mathcal{N}$, and for two distinct agents i and j, $W_{ij} > 0$ if and only if $(i,j) \in \mathcal{E}$.

When Assumption 1 is satisfied, we have [12]

$$\rho := \left\|W - n^{-1}\mathbf{1}_n\mathbf{1}_n^\top\right\| < 1. \tag{4}$$

We present two consensus-based algorithms that will serve as the basis for designing distributed zero-order algorithms.

1) The *DGD* algorithm [3], [10] is given by the following iterations:

$$x^i(t) = \sum_{j=1}^n W_{ij}x^j(t-1) - \eta_t\nabla f_i(x^i(t-1)) \tag{5}$$

where $x^i(t) \in \mathbb{R}^d$ denotes the local copy of the decision variable for the ith agent, and η_t is the step size. It has been shown that DGD in general converges more slowly than the centralized gradient descent algorithm [3], [12] for smooth functions. This is because the local gradient ∇f_i does not vanish at the stationary point, and a diminishing step size η_t is necessary, which slows down the convergence.

2) The *DGD gradient tracking* method incorporates additional local variables $s^i(t)$ to track the global gradient $\nabla f = \frac{1}{n}\sum_i \nabla f_i$

$$s^i(t) = \sum_{j=1}^n W_{ij}s^j(t-1) + \nabla f_i(x^i(t-1)) - \nabla f_i(x^i(t-2))$$

$$x^i(t) = \sum_{j=1}^n W_{ij}x^j(t-1) - \eta_t s^i(t)$$

where we set $s^i(1) = \nabla f_i(x^i(0))$ for each i. Since gradient tracking has been proposed, it has attracted much attention and inspired many recent studies [7], [8], [12], [15], [20], as it can accelerate the convergence for smooth objectives compared to DGD. Here, we provide a high-level explanation of how gradient tracking works: For smooth functions, when $x^i(t)$ approaches consensus, $\nabla f_i(x^i(t))$ will not change much because of the smoothness and, therefore, the local variables $s^i(t)$ will eventually reach a consensus; on the other hand, by induction, it can be shown that

$$\frac{1}{n}\sum_{i=1}^n s^i(t) = \frac{1}{n}\sum_{i=1}^n \nabla f_i(x^i(t-1)).$$

Therefore, the sequence $(s^i(t))_{t\geq 1}$ will eventually converge to the global gradient, and a constant step size $\eta_t = \eta$ is allowed, leading to comparable convergence rates as the centralized gradient methods. See [12, Sec. III and IV.B] for more discussion.

Algorithm 1: 2-Point Gradient Estimator Without Global Gradient Tracking.

for $t = 1, 2, 3, \ldots$ **do**
  foreach $i \in \mathcal{N}$ **do**
    1) Generate $z^i(t) \sim \mathcal{U}(\mathbb{S}_{d-1})$ independently from $(z^i(\tau))_{\tau=1}^{t-1}$ and $(z^j(\tau))_{\tau=1}^{t}$ for $j \neq i$.;
    2) Update $x^i(t)$ by

$$g^i(t) = \mathsf{G}_{f_i}^{(2)}(x^i(t-1); u_t, z^i(t)), \tag{6}$$

$$x^i(t) = \sum_{j=1}^n W_{ij}(x^j(t-1) - \eta_t g^j(t)). \tag{7}$$

  end
end

C. Our Algorithms

Following the previous discussions, it would be ideal if we can combine the 2-point gradient estimator and the DGD with gradient tracking and maintain a convergence rate comparable to the centralized vanilla gradient descent method. However, it turns out that such combination does not lead to the desired convergence rate. This is mainly because gradient tracking requires increasingly accurate local gradient information as one approaches the stationary point to achieve faster convergence compared to DGD, whereas the 2-point gradient estimator can produce a variance that does not decrease to zero even if the radius u decreases to zero; a more detailed explanation will be provided in Section III-C.

We propose the following two distributed zero-order algorithms for the problem (1).[1]

1) Algorithm 1 employs the 2-point gradient estimator (3), and adopts the consensus procedure of the DGD

[1]For both algorithms, we employ the *adapt-then-combine* (ATC) strategy [33], a commonly used variant for consensus optimization that is slightly different from the combine-then-adapt (CTA) strategy in (5). Both ATC and CTA can be used in our algorithms, and the convergence results will be similar.

Algorithm 2: $2d$-Point Gradient Estimator With Global Gradient Tracking.

Set $s^i(0) = g^i(0) = 0$ for each $i \in \mathcal{N}$.
for $t = 1, 2, 3, \ldots$ **do**
  foreach $i \in \mathcal{N}$ **do**
    1) Update $s^i(t)$ by

$$g^i(t) = \mathsf{G}_{f_i}^{(2d)}(x^i(t-1); u_t), \tag{8}$$

$$s^i(t) = \sum_{j=1}^{n} W_{ij}\big(s^j(t-1) + g^j(t) - g^j(t-1)\big). \tag{9}$$

    2) Update $x^i(t)$ by

$$x^i(t) = \sum_{j=1}^{n} W_{ij}(x^j(t-1) - \eta s^j(t)). \tag{10}$$

  end
end

algorithm that only involves averaging over the local decision variables.

2) Algorithm 2 employs the $2d$-point gradient estimator (2), and adopts the consensus procedure of the gradient tracking method where the auxiliary variable $s^i(t)$ is introduced to track the global gradient $\nabla f = \frac{1}{n}\sum_i \nabla f_i$. We shall see in Theorems 3 and 4 that $s^i(t)$ converges to the gradient of the global objective function as $t \to \infty$ under mild conditions.

III. Main Results

In this section, we present the convergence results of our algorithms. Due to space limit, we only provide proof sketches in the main text (Theorem 1) and in the Appendix (Theor. 2, 3, and 4) and refer to [34] for complete proofs.

A. Convergence of Algorithm 1

Let $x^i(t)$ denote the sequence generated by Algorithm 1 with a positive, nonincreasing sequence of step sizes η_t. Denote

$$\bar{x}(t) := \frac{1}{n}\sum_{i=1}^{n} x^i(t), \qquad R_0 := \frac{1}{n}\sum_{i=1}^{n} \|x^i(0) - \bar{x}(0)\|^2.$$

We first analyze the case with general nonconvex smooth objective functions.

Theorem 1: Assume that each local objective function f_i is uniformly G-Lipschitz and L-smooth for some positive constants G and L, and that $f^* := \inf_{x\in\mathbb{R}^d} f(x) > -\infty$.

1) Suppose $\eta_1 L \le 1/4$, $\sum_{t=1}^{\infty}\eta_t = +\infty$, $\sum_{t=1}^{\infty}\eta_t^2 < +\infty$, and $\sum_{t=1}^{\infty}\eta_t u_t^2 < +\infty$. Then almost surely, $\|x^i(t) - \bar{x}(t)\|$ converges to zero for all $i \in \mathcal{N}$, $\nabla f(\bar{x}(t))$ converges to zero, and $\lim_{t\to\infty} f(\bar{x}(t))$ exists.
2) Suppose that

$$\eta_t = \frac{\alpha_\eta}{4L\sqrt{d}}\cdot\frac{1}{\sqrt{t}}, \qquad u_t \le \frac{\alpha_u G}{L\sqrt{d}}\cdot\frac{1}{t^{\gamma/2-1/4}}$$

with $\alpha_\eta \in (0, 1]$, $\alpha_u \ge 0$, and $\gamma > 1$. Then almost surely, $\|x^i(t) - \bar{x}(t)\|$ converges to zero for all i, and $\liminf_{t\to\infty}\|\nabla f(\bar{x}(t))\| = 0$. Furthermore, we have

$$\frac{\sum_{\tau=0}^{t-1}\eta_{\tau+1}\mathbb{E}\big[\|\nabla f(\bar{x}(\tau))\|^2\big]}{\sum_{\tau=0}^{t-1}\eta_{\tau+1}} \le \sqrt{\frac{d}{t}}\left[\frac{\alpha_\eta G^2}{3n}\ln(2t+1) + \frac{8L(f(\bar{x}(0))-f^*)}{\alpha_\eta} + \frac{6R_0\,L^2}{(1-\rho^2)\sqrt{d}} + \frac{9\alpha_\eta^2\kappa^2\rho^2\,G^2}{(1-\rho^2)^2\sqrt{d}} + \frac{9\alpha_u^2\gamma G^2}{4(\gamma-1)}\right] + o\left(\frac{1}{\sqrt{t}}\right) \tag{11}$$

where κ is some positive numerical constant, and

$$\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\big[\|x^i(t) - \bar{x}(t)\|^2\big] \le \frac{\alpha_\eta^2\kappa^2\rho^2}{(1-\rho^2)^2}\frac{G^2/L^2}{t} + o(t^{-1}). \tag{12}$$

Proof Sketch of Theorem 1: We shall only provide a proof sketch of the bounds (11) and (12). Denote

$$x(t) := \begin{bmatrix} x^1(t)^\top & \cdots & x^n(t)^\top \end{bmatrix}^\top$$
$$g(t) := \begin{bmatrix} g^1(t)^\top & \cdots & g^n(t)^\top \end{bmatrix}^\top$$

and $\bar{g}(t) := \frac{1}{n}\sum_{i=1}^{n} g^i(t)$, $\delta(t) := f(\bar{x}(t)) - f^*$, $e_c(t) := \mathbb{E}[\|x(t) - 1_n\otimes\bar{x}(t)\|^2]$.

The proof relies on three lemmas. The first lemma analyzes how the objective value at the averaged iterate $f(\bar{x}(t))$ evolves as the iterations proceed. Its proof is based on the L-smoothness of the function f and Proposition 1.

Lemma 1: We have

$$\mathbb{E}[f(\bar{x}(t))] \le \mathbb{E}[f(\bar{x}(t-1))] - \frac{\eta_t}{2}\mathbb{E}\big[\|\nabla f(\bar{x}(t-1))\|^2\big] + \frac{\eta_t L^2}{n}e_c(t-1) + \frac{\eta_t^2 L}{2}\mathbb{E}\big[\|\bar{g}(t)\|^2\big] + \eta_t u_t^2\,L^2. \tag{13}$$

This lemma suggests that we further need to bound two terms, the second moment of $\bar{g}(t)$ and the expected consensus error $e_c(t-1)$. This is tackled by the following two lemmas.

Lemma 2: We have

$$\mathbb{E}\big[\|\bar{g}(t)\|^2\big] \le \frac{4\,G^2 d}{3n} + 2\mathbb{E}\big[\|\nabla f(\bar{x}(t-1))\|^2\big] + \frac{4\,L^2}{n}e_c(t-1) + u_t^2\,L^2\,d^2.$$

Lemma 3: We have

$$e_c(t) \le \left(\frac{1+\rho^2}{2}\right)^t e_c(0) + \frac{8n\rho^2\kappa^2}{1-\rho^2}G^2\,d\sum_{\tau=0}^{t-1}\left(\frac{1+\rho^2}{2}\right)^\tau \eta_{t-\tau}^2 \tag{14}$$

where $\kappa > 0$ is some numerical constant.

We mention that Lemmas 2 and 3 are based on [25, Lemma 10] and a standard result in consensus optimization (see Appendix A).

Now, by plugging the bound of Lemma 2 into (13) and noticing that $\eta_t L \le 1/4$, we get

$$\mathbb{E}[\delta(t)] \le \mathbb{E}[\delta(t-1)] - \frac{\eta_t}{4}\|\nabla f(\bar{x}(t-1))\|^2 + \frac{3\eta_t L^2}{2n}e_c(t-1) + \frac{2\eta_t^2 LG^2\,d}{3n} + \eta_t u_t^2\,L^2\left(1 + \frac{1}{2}d^2\eta_t L\right). \tag{15}$$

By telescoping sum and noting that $\delta(t) \geq 0$, we get

$$\sum_{\tau=1}^{t} \eta_\tau \mathbb{E}\left[\|\nabla f(\bar{x}(t-1))\|^2\right] \leq 4\delta(0) + \frac{6\,L^2}{n}\sum_{\tau=1}^{t}\eta_\tau e_{\rm c}(\tau-1) + \frac{8LG^2\,d}{3n}\sum_{\tau=1}^{t}\eta_\tau^2 + 4\,L^2\sum_{\tau=1}^{t}\left(\eta_\tau u_\tau^2 + \frac{1}{2}d^2\,L\eta_\tau^2 u_\tau^2\right). \tag{16}$$

Since $\eta_t = \alpha_\eta/(4L\sqrt{d\cdot t})$ and $u_t \leq \alpha_u G/(L\sqrt{d}t^{\gamma/2-1/4})$ with $\alpha_\eta \leq 1$ and $\gamma > 1$, it can be shown that

$$\sum_{\tau=1}^{t}\eta_\tau \geq 2\eta_1(\sqrt{t+1}-1),\ \sum_{\tau=1}^{t}\eta_\tau^2 \leq \eta_1^2\ln(2t+1)$$

$$\sum_{\tau=1}^{t}\left(\eta_\tau u_\tau^2 + \frac{1}{2}d^2L\eta_\tau^2u_\tau^2\right) \leq \eta_1\frac{9\alpha_u^2\,G^2\sqrt{d}}{8\,L^2}\frac{\gamma}{\gamma-1}$$

and by Lemma 3, we can show that

$$\begin{aligned}\sum_{\tau=1}^{t}\eta_\tau\frac{e_{\rm c}(\tau-1)}{n} &\leq \eta_1\frac{e_{\rm c}(0)}{n}\sum_{t=1}^{\infty}\left(\frac{1+\rho^2}{2}\right)^{t-1} + \eta_1^3\frac{8\rho^2\kappa^2}{1-\rho^2}G^2\,d \\ &\quad\times\sum_{t=2}^{\infty}\sum_{\tau=0}^{t-2}\frac{1}{\sqrt{t}(t-1-\tau)}\left(\frac{1+\rho^2}{2}\right)^{\tau} \\ &\leq \frac{2\eta_1 e_{\rm c}(0)}{n(1-\rho^2)} + \eta_1^3\frac{48\kappa^2\rho^2}{(1-\rho^2)^2}G^2\,d.\end{aligned}$$

By plugging these bounds into (16), we get the bound (11). The bound (12) is a direct consequence of Lemma 3 and the fact that $\sum_{\tau=0}^{t-1}\lambda^\tau/(t-\tau) = ((1-\lambda)t)^{-1} + o(t^{-1})$. ■

Remark 1: Note that in (11), we use the squared norm of the gradient to assess the suboptimality of the iterates, and characterize the convergence by *ergodic* rates. This type of convergence rate bound is common for local methods of unconstrained nonconvex problems where we do not aim for global optimal solutions [9], [35].

Remark 2: Each iteration of Algorithm 1 requires two queries of function values. Thus, the convergence rate (11) can also be interpreted as $O(\sqrt{d/m}\log m)$ where m denotes the number of function value queries. Characterizing convergence rate in terms of the number of function value queries m and the dimension d is conventional for zero-order optimization. In scenarios where zero-order methods are applied, the computation of the function values is usually one of the most time-consuming procedures. In addition, it is also of interest to characterize how the convergence scales with the dimension d.

The following theorem shows that for a gradient dominated global objective, a better convergence rate can be achieved.

Theorem 2: Assume that each local objective function f_i is uniformly L-smooth for some $L > 0$. Furthermore, assume that $\inf_{x\in\mathbb{R}^d} f_i(x) = f_i^* > -\infty$ for each i, and that the global objective function f is μ-gradient dominated and has a minimum value denoted by f^*. Suppose

$$\eta_t = \frac{2\alpha_\eta}{\mu(t+t_0)},\quad u_t \leq \frac{\alpha_u}{\sqrt{t+t_0}}$$

for some $\alpha_\eta > 1$ and $\alpha_u > 0$, where

$$t_0 \geq \frac{2\alpha_\eta L}{\mu(1-\rho^2)}\left(\frac{32Ld}{3\mu} + 9\rho\right) - 1.$$

Then, using Algorithm 1, we have

$$\mathbb{E}[f(\bar{x}(t)) - f^*] \leq \left(\frac{32\alpha_\eta^2\,L^2\Delta}{\mu^2} + \frac{6\alpha_\eta\alpha_u^2\,L^2}{\mu}\right)\frac{d}{t} + o(t^{-1}) \tag{17}$$

and

$$\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[\|x^i(t) - \bar{x}(t)\|^2\right] \leq \frac{32\alpha_\eta^2\rho^2L\Delta}{\mu^2(1-\rho^2)}\left(\frac{4\,d}{3} + \frac{6\rho^2}{1-\rho^2}\right)\frac{1}{t^2} + o(t^{-2}) \tag{18}$$

where $\Delta := f^* - \frac{1}{n}\sum_{i=1}^{n} f_i^*$.

Remark 3: The convergence rate (17) can also be described as $\mathbb{E}[f(\bar{x}(t)) - f^*] = O(d/m)$, where m is the number of function value queries.

Table I summarizes that, while Algorithm 1 employs a randomized 2-point zero-order estimator of ∇f_i, its convergence rates are comparable with the DGD algorithm [6], [36]. However, its convergence rates are inferior to its centralized zero-order counterpart in [9].

B. Convergence of Algorithm 2

Let $(x^i(t), s^i(t))$ denote the sequence generated by Algorithm 2 with a constant step size η. Denote

$$\begin{aligned}\bar{x}(t) &:= \frac{1}{n}\sum_{i=1}^{n}x^i(t) \\ R_0 &:= \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\eta\rho^2}{2\,L}\|\nabla f_i(x^i(0))\|^2 + \|x^i(0) - \bar{x}(0)\|^2\right) \\ &\quad + \frac{\eta\rho^2u_1^2Ld}{4}.\end{aligned}$$

We first analyze the case where the local objectives are nonconvex and smooth.

Theorem 3: Assume that each local objective function f_i is uniformly L-smooth for some positive constant L, and that $f^* := \inf_{x\in\mathbb{R}^d} f(x) > -\infty$. Suppose

$$\eta L \leq \min\left\{\frac{1}{6}, \frac{(1-\rho^2)^2}{4\rho^2(3+4\rho^2)}\right\},\quad R_u := d\sum_{t=1}^{\infty}u_t^2 < +\infty$$

and that u_t is nonincreasing. Then, $\lim_{t\to\infty} f(\bar{x}(t))$ exists

$$\frac{1}{t}\sum_{\tau=0}^{t-1}\|\nabla f(\bar{x}(\tau))\|^2 \leq \frac{1}{t}\left[\frac{3.2(f(\bar{x}(0)) - f^*)}{\eta} + \frac{12.8\,L^2R_0}{1-\rho^2} + 2.4R_uL^2\right] \tag{19}$$

and

$$\frac{1}{t}\sum_{\tau=0}^{t-1}\frac{1}{n}\sum_{i=1}^{n}\|x^i(\tau)-\bar{x}(\tau)\|^2 \le \frac{1}{t}\left[1.6\eta(f(\bar{x}(0))-f^*)+\frac{3.2R_0}{1-\rho^2}+0.35R_u\right] \tag{20}$$

$$\frac{1}{t}\sum_{\tau=1}^{t}\frac{1}{n}\sum_{i=1}^{n}\|s^i(\tau)-\nabla f(\bar{x}(\tau-1))\|^2 \le \frac{1}{t}\left[9.6L(f(\bar{x}(0))-f^*)+\frac{19.2\,LR_0}{\eta(1-\rho^2)}+\frac{2.35}{\eta}LR_u\right]. \tag{21}$$

Remark 4: Theorem 3 shows that Algorithm 2 achieves a convergence rate of $O(1/t)$ in terms of the averaged squared norm of $\nabla f(\bar{x}(t))$, and has a consensus rate of $O(1/t)$ for the averages of the squared consensus error $\|x^i(t)-\bar{x}(t)\|^2$ and the squared gradient tracking error $\|s^i(t)-\nabla f(\bar{x}(t-1))\|^2$. They match the rates for distributed nonconvex optimization with gradient tracking [7]. On the other hand, since each iteration requires $2d$ queries of function values, we get a $O(d/m)$ rate in terms of the number of function value queries m. This matches the convergence rate of centralized zero-order algorithms without Nesterov-type acceleration [9].

Now we proceed to the situation with a gradient dominated global objective.

Theorem 4: Assume that each local objective function f_i is uniformly L-smooth for some positive constant L, and that the global objective function f is μ-gradient dominated and achieves its global minimum at x^*. Suppose the step size η satisfies

$$\eta L=\alpha\cdot\left(\frac{\mu}{L}\right)^{\frac{1}{3}}\frac{(1-\rho^2)^2}{14} \tag{22}$$

for some $\alpha\in(0,1]$, and $(u_t)_{t\ge1}$ is nonincreasing. Let

$$\lambda:=1-\alpha\left(\frac{1-\rho^2}{5}\right)^2\left(\frac{\mu}{L}\right)^{\frac{4}{3}}.$$

Then

$$f(\bar{x}(t))-f(x^*)\le O(\lambda^t)+5(1-\rho^2)Ld\sum_{\tau=0}^{t-1}\lambda^\tau u_{t-\tau}^2 \tag{23}$$

$$\frac{1}{n}\sum_{i=1}^{n}\|x^i(t)-\bar{x}(t)\|^2\le O(\lambda^t)+\frac{3\eta Ld}{1-\rho^2}\sum_{\tau=0}^{t-1}\lambda^\tau u_{t-\tau}^2 \tag{24}$$

$$\frac{1}{n}\sum_{i=1}^{n}\|s^i(t)-\nabla f(\bar{x}(t-1))\|^2\le O(\lambda^t)+\frac{18\,L^2\,d}{1-\rho^2}\sum_{\tau=0}^{t-1}\lambda^\tau u_{t-\tau}^2. \tag{25}$$

Remark 5: If we use an exponentially decreasing sequence $u_t\propto\tilde{\lambda}^{t/2}$ with $\tilde{\lambda}<\lambda$, then both the objective error $f(\bar{x}(t))-f(x^*)$ and the consensus errors $\|x^i(t)-\bar{x}(t)\|^2$ and $\|s^i(t)-\nabla f(\bar{x}(t-1))\|^2$ achieve linear convergence rate $O(\lambda^t)$, or $O(\lambda^{m/d})$ in terms of the number of function value queries. In addition, we notice that the decaying factor λ given by Theorem 4 has a better dependence on μ/L than in [8] for convex problems. We point out that this is not a result of using zero-order techniques, but rather a more refined analysis of the gradient tracking procedure.

Remark 6: Note that the conditions on the step sizes in Theorems 2–4 depend on ρ, a measure of the connectivity of the network. In order to choose step sizes to satisfy these conditions in the distributed setting, one possible approach is as follows: Assuming that each agent knows an upper bound $\overline{n}$ on the total number of agents, by [37, Lemma 2], if one chooses W to be the lazy Metropolis matrix, then $\rho\le1-1/(71\overline{n}^2)$, based on which the agents can then derive their step sizes according to the conditions in the theorems. We also note that some existing works (e.g., [38]) attempt to get rid of the dependence of step sizes on the graph topology, and whether those techniques can be applied in our work is beyond the scope of this article but is an interesting future direction.

C. Comparison of the Two Algorithms

We see from the above results that Algorithm 2 converges faster than Algorithm 1 asymptotically as $m\to\infty$ in theory. However, each iteration of Algorithm 2 makes progress only after $2d$ queries of function values, which could be an issue if d is very large. On the contrary, each iteration of Algorithm 1 only requires two function value queries, meaning that progress can be made relatively immediately without exploring all the d dimensions. This observation suggests that, when neglecting communication delays, Algorithm 1 is more favorable for high-dimensional problems, whereas Algorithm 2 could handle problems of relatively low dimensions better with faster convergence.

We emphasize that there still exists a tradeoff between the convergence rate and the ability to handle high-dimensional problems even if one combines the 2-point gradient estimator (2) with the gradient tracking method as

$$\begin{aligned} g^i(t)&=\mathsf{G}_{f_i}^{(2)}(x^i(t-1);u_t,z^i(t)),\quad z^i(t)\sim\mathcal{U}(\mathbb{S}_{d-1})\\ s^i(t)&=\sum_{j=1}^{n}W_{ij}\big(s^j(t-1)+g^j(t)-g^j(t-1)\big)\\ x^i(t)&=\sum_{j=1}^{n}W_{ij}(x^j(t-1)-\eta s^j(t)). \end{aligned} \tag{26}$$

Theoretical analysis suggests that, in order for $s^i(t)$ to reach a consensus in the sense that $\mathbb{E}[\|s^i(t)-s^j(t)\|^2]\to0$, we need

$$\lim_{t\to\infty}\mathbb{E}\left[\|g^i(t)-g^i(t-1)\|^2\right]\to0.$$

On the other hand, we have the following lemma regarding the variance of the 2-point gradient estimator $\mathsf{G}_f^{(2)}(x;u,z)$.

Lemma 4: Let $f:\mathbb{R}^d\to\mathbb{R}$ be an arbitrary L-smooth function. Then

$$\lim_{u\to0^+}\mathbb{E}_z\left[\|\mathsf{G}_f^{(2)}(x;u,z)-\nabla f^u(x)\|^2\right]=(d-1)\|\nabla f(x)\|^2$$

where $z\sim\mathcal{U}(\mathbb{S}_{d-1})$.

Proof: Notice that for any $z \in \mathbb{S}_{d-1}$ and $u \in (0, 1]$, we have

$$\left|\frac{f(x+uz) - f(x-uz)}{2\,u}\right| \leq \sup_{y\in\mathbb{B}_d} \|\nabla f(x+y)\|.$$

Therefore

$$\begin{aligned}
&\lim_{u\to 0} \mathbb{E}_z\left[\|\mathsf{G}_f^{(2)}(x;u,z)\|^2\right] \\
&= d^2\mathbb{E}_z\left[\left|\lim_{u\to 0}\frac{f(x+uz)-f(x-uz)}{2\,u}\right|^2\right] = d^2\mathbb{E}_z\left[\left|\nabla f(x)^\top z\right|^2\right] \\
&= d^2\nabla f(x)^\top \mathbb{E}_z\left[zz^\top\right]\nabla f(x) = d\|\nabla f(x)\|^2
\end{aligned}$$

where in the second step we exchanged the order of limit and expectation by the bounded convergence theorem, and in the last step we used $d\,\mathbb{E}_z[zz^\top] = I_d$ for $z \sim \mathcal{U}(\mathbb{S}_{d-1})$. Then, noticing that $\nabla f^u(x) \to \nabla f(x)$ as $u \to 0$, we get

$$\begin{aligned}
&\lim_{u\to 0} \mathbb{E}_z\left[\|\mathsf{G}_f^{(2)}(x;u,z) - \nabla f^u(x)\|^2\right] \\
&= \lim_{u\to 0}\left(\mathbb{E}_z\left[\|\mathsf{G}_f^{(2)}(x;u,z)\|^2\right] - \|\nabla f^u(x)\|^2\right) \\
&= (d-1)\|\nabla f(x)\|^2.
\end{aligned}$$

■

Lemma 4 suggests that each gradient estimator $\mathsf{G}_{f_i}^{(2)}(x^i(t-1); u_t, z^i(t))$ in (26) will produce a nonvanishing variance approximately equal to $(d-1)\,\mathbb{E}[\|\nabla f_i(x^i(t-1))\|^2]$ even if we let $u \to 0$ as $x^i(t)$ approaches a stationary point. Consequently, $\mathbb{E}[\|g^i(t) - g^i(t-1)\|^2]$ is not guaranteed to converge to zero as $t \to \infty$. The nonvanishing variance will then be reflected in $s^i(t)$ that tracks the global gradient, and consequently the overall convergence will be slowed down. We refer to [8], [12], and [39] for related analysis, and to Section IV for a numerical example.

D. Comparison With Existing Algorithms

In this section, we provide a detailed comparison with existing literature on distributed zero-order optimization, specifically [2], [26], and [27].

1) Papers [26] and [27] discuss convex problems, whereas paper [2] and our work focus on nonconvex problems.
2) In terms of the assumptions on the noisy function queries, paper [27] and our work consider a noise-free case. Paper [2] considers stochastic queries but assumes two function values can be obtained for a single random sample. Paper [26] assumes independent additive noise on each function value query. We expect that our Algorithm 1 can be generalized to the setting adopted in [2] with heavier mathematics. Extensions to general stochastic cases remain our ongoing work.
3) In terms of the approach to reach consensus among agents, our algorithms are similar to [26] and [27], where some weighted average of the neighbors' local variables is utilized, whereas paper [2] uses the method of multipliers to design their algorithms. We also mention that our Algorithm 2 employs the gradient tracking technique, which, to our best knowledge, has not been discussed in existing literature on distributed zero-order optimization yet.
4) Regarding the convergence rates for nonconvex optimization, paper [2] proved that its proposed ZONE algorithm achieves $O(1/T)$ rate if each iteration also employs $O(T)$ function value queries, where T is the number of iterations planned in advance. Therefore, in terms of the number of function value queries M, its convergence rate is in fact $O(1/\sqrt{M})$, which is roughly comparable with Algorithm 1 and slower than Algorithm 2 in our article. Also, paper [2] did not discuss the dependence on the problem dimension d. Moreover, our algorithms only require constant numbers (2 or $2d$) of function value queries which is more appealing for practical implementation when T is set to be very large for achieving sufficiently accurate solutions.

IV. Numerical Examples

We consider a multiagent nonconvex optimization problem formulated as

$$\begin{aligned}
&\min_{x\in\mathbb{R}^d}\ \frac{1}{n}\sum_{i=1}^n f_i(x) \\
&f_i(x) = \frac{a_i}{1+\exp(-\xi_i^\top x - \nu_i)} + b_i\ln(1+\|x\|^2)
\end{aligned} \tag{27}$$

where $a_i, b_i, \nu_i \in \mathbb{R}$ and $\xi_i \in \mathbb{R}^d$ for each $i = 1, \ldots, N$.

For the numerical example, we set the dimension to be $d = 64$ and the number of agents to be $n = 50$. The parameters a_i, ν_i and each entry of ξ_i are randomly generated from the standard Gaussian distribution, and $(b_1, \ldots, b_n)$ is generated from the distribution $\mathcal{N}(1_n, I_n - \frac{1}{n}1_n 1_n^\top)$ so that $\frac{1}{n}\sum_i b_i = 1$. The graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ is generated by uniformly randomly sampling n points on $\mathbb{S}_2$, and then connecting pairs of points with spherical distances less than $\pi/4$. The Metropolis–Hastings weights [40] are employed for constructing W.

We compare the following algorithms on the problem (27):

1) Algorithm 1 with $\eta_t = 0.02/\sqrt{t}$ and $u_t = 4/\sqrt{t}$.
2) Algorithm 2 with $\eta = 0.02$ and $u_t = 4/t^{3/4}$.
3) ZONE-M [2], where we test two setups $J = 1$, $\rho_t = 4\sqrt{t}$, $u_t = 4/\sqrt{t}$ and $J = 100$, $\rho_t = 0.4\sqrt{t}$, $u_t = 4/\sqrt{t}$.
4) 2-point gradient estimator combined with gradient tracking [see (26)] with $\eta = 2\times 10^{-4}$ and $u_t = 4/t^{3/4}$.

All algorithms start from the same initial points, which are randomly generated from the distribution $\mathcal{N}(0, \frac{25}{d}I_d)$ for each agent.

A. Comparison of Algorithms 1 and 2

Fig. 1 shows the convergence behavior of Algorithms 1 and 2, where the top figure illustrates the squared norm of the gradient at $\bar{x}(t)$, and the bottom figure illustrates the consensus error $\frac{1}{n}\sum_{i=1}^n \|x^i(t) - \bar{x}(t)\|^2$. The horizontal axis has been normalized as the number of function value queries m. We can see that Algorithm 1 converges faster during the initial stage, but then slows down and converges at a relatively stable sublinear rate. The convergence of Algorithm 2 is relatively slow initially, but

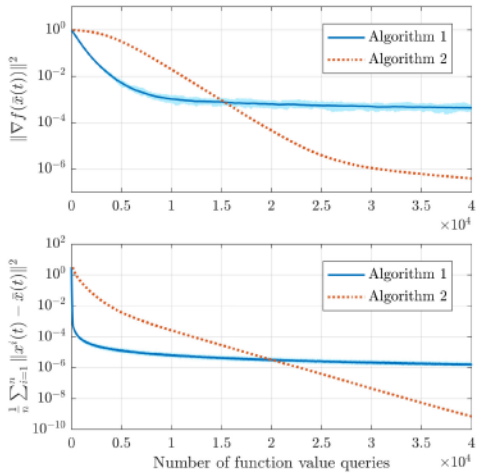


Fig. 1. Convergence of Algorithms 1 and 2. For Algorithm 1, the light blue shaded areas represent the results for 50 random instances, and the dark blue curves represent their average.

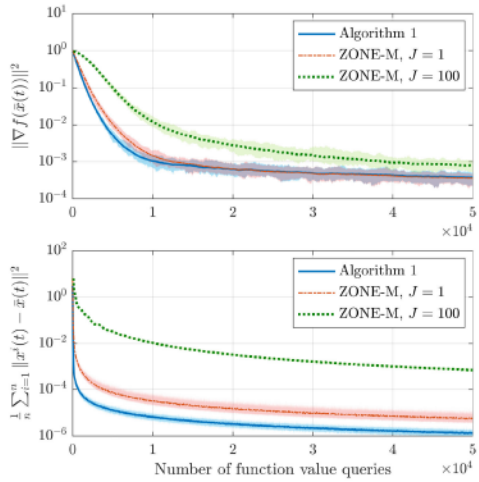


Fig. 2. Convergence of Algorithm 1 and ZONE-M with $J = 1$ and $J = 100$. For each algorithm, the light shaded areas represent the results for 50 random instances, and the dark curves represent their average.

then becomes faster as $m \gtrsim 0.5 \times 10^4$, and when $m \gtrsim 2 \times 10^4$, Algorithm 2 achieves smaller squared gradient norm and consensus error compared to Algorithm 1; the convergence slows down as m exceeds 2.5×10^4 but is still faster than Algorithm 1. Further investigation of the simulation results suggests that the speed-up of Algorithm 2 within $0.5 \times 10^4 \lesssim m \lesssim 2.5 \times 10^4$ is due to $\bar{x}(t)$ becoming sufficiently close to a local optimal, around which the objective function is locally strongly convex; the slow-down after m exceeds 2.5×10^4 is caused by the zero-order gradient estimation error that becomes dominant, and can be postponed or avoided if we let u_t decrease more aggressively.

From these results, it can be seen that, if the total number of function value queries is limited by, say $m \lesssim 1.5 \times 10^4$, then Algorithm 1 might be favorable compared to Algorithm 2 despite slower asymptotic convergence rate, whereas if more function value queries are allowed, then Algorithm 2 could be favored. We observe that this is related with the discussion in Section III-C.

B. Comparison With Other Algorithms

Fig. 2 compares the convergence of Algorithm 1 and the two setups of ZONE-M, including the curves for the squared norm of the gradient $\|\nabla f(\bar{x}(t))\|^2$ and the consensus error $\frac{1}{n}\sum_{i=1}^n \|x^i(t) - \bar{x}(t)\|^2$. The horizontal axis has been normalized as the number of function value queries m. It can be seen that Algorithm 1 and ZONE-M with $\rho_t \propto \sqrt{t}$, $J = 1$ have similar convergence behavior. For ZONE-M with $\rho_t \propto \sqrt{t}$ and $J = 100$, while the convergence of $\|\nabla f(\bar{x}(t))\|^2$ is comparable with Algorithm 1 and ZONE-M with $J = 1$, the consensus error decreases much more slowly, as ZONE-M with $J = 100$ conducts much fewer consensus averaging steps per function value query compared to Algorithm 1 and ZONE-M with $J = 1$.

Fig. 3 compares the convergence of Algorithm 2 and the 2-point estimator combined with gradient tracking (26), including the curves for the squared norm of the gradient $\|\nabla f(\bar{x}(t))\|^2$, the consensus error $\frac{1}{n}\sum_{i=1}^n \|x^i(t) - \bar{x}(t)\|^2$, and also the gradient

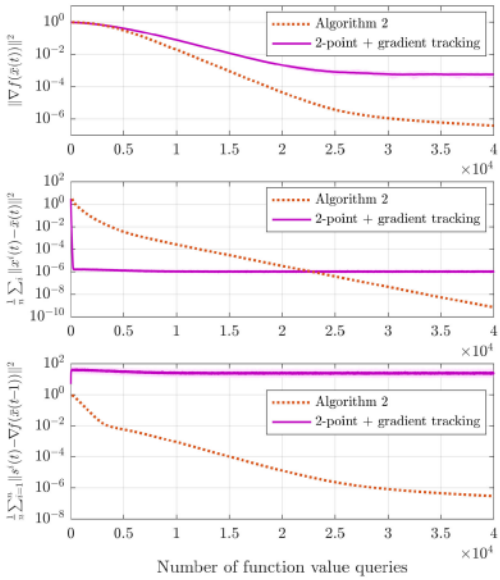


Fig. 3. Convergence of Algorithm 2 and 2-point estimator combined with gradient tracking. For 2-point estimator combined with gradient tracking, the light pink shaded areas represent the results for 50 random instances, and the dark purple curves represent their average.

tracking error $\frac{1}{n}\sum_{i=1}^n \|s^i(t) - \nabla f(\bar{x}(t-1))\|^2$. It is straightforward to see that Algorithm 2 has better asymptotic convergence behavior than the 2-point estimator combined with gradient tracking. Moreover, for the 2-point estimator combined with gradient tracking, the gradient tracking error does not converge to zero but remains at a constant level, indicating that the gradient tracking technique is ineffective in this case. These observations are in accordance with our theoretical discussion in Section III-C.

V. Conclusion

We proposed two distributed zero-order algorithms for nonconvex multiagent optimization, established theoretical results

on their convergence rates, and showed that they achieve comparable performance with their distributed gradient-based or centralized zero-order counterparts. We also provided a brief discussion on how the dimension of the problem will affect their performance in practice. There are many lines of future work, such as follows:

1) introducing noise or errors when evaluating $f_i(x)$;
2) investigating how to escape from saddle point for distributed zero-order methods;
3) extension to nonsmooth problems;
4) investigating whether the step sizes can be independent of the network topology;
5) studying time-varying graphs;
6) investigating the fundamental gap between centralized methods and distributed methods, especially for high-dimensional problems.

APPENDIX A
AUXILIARY RESULTS AND NOTATIONS

The following lemma is a standard result in consensus optimization [12].

Lemma 5: Let ρ be defined by (4). Then, for any $x^1, \ldots, x^n \in \mathbb{R}^d$, we have

$$\|(W \otimes I_d)(x - \mathbf{1}_n \otimes \bar{x})\| \leq \rho \|x - \mathbf{1}_n \otimes \bar{x}\|$$

where $x = \begin{bmatrix} x_1^\top & \cdots & x_n^\top \end{bmatrix}^\top \in \mathbb{R}^{nd}$, $\bar{x} = \frac{1}{n}\sum_{i=1}^n x^i$.

We will use the notations

$$x(t) := \begin{bmatrix} x^1(t)^\top & \cdots & x^n(t)^\top \end{bmatrix}^\top$$

$$g(t) := \begin{bmatrix} g^1(t)^\top & \cdots & g^n(t)^\top \end{bmatrix}^\top$$

and $\bar{x}(t) := \frac{1}{n}\sum_{i=1}^n x^i(t)$, $\bar{g}(t) := \frac{1}{n}\sum_{i=1}^n g^i(t)$, $\delta(t) := f(\bar{x}(t)) - f^*$ for all subsequent analysis of Algorithms 1 and 2.

APPENDIX B
PROOF SKETCH OF THEOREM 2

Let $e_c(t) := \mathbb{E}[\|x(t) - \mathbf{1}_n \otimes \bar{x}(t)\|^2]$. We recall that each f_i is L-smooth and is lower bounded by f_i^* over $\mathbb{R}^d$, and that $\Delta := f^* - \frac{1}{n}\sum_{i=1}^n f_i^*$.

Note that Lemma 1 still applies here. On the other hand, as each f_i is not uniformly Lipschitz continuous over $\mathbb{R}^d$, we need new lemmas characterizing the consensus procedure and the second moment of $\bar{g}(t)$.

Lemma 6: Suppose the step sizes satisfy the condition of Theorem 2. Then for each $t \geq 1$, we have

$$\frac{e_c(t)}{n} \leq \frac{1+\rho^2}{2}\frac{e_c(t-1)}{n} + 4\eta_t^2\rho^2 L\left(\frac{4}{3}\frac{d}{} + \frac{6\rho^2}{1-\rho^2}\right)\mathbb{E}[\delta(t-1)] + 4\eta_t^2\rho^2 L\Delta\left(\frac{4}{3}\frac{d}{} + \frac{6\rho^2}{1-\rho^2}\right) + \eta_t^2\rho^2 u_t^2 L^2\left(d^2 + \frac{6\rho^2}{1-\rho^2}\right). \tag{28}$$

Lemma 7: We have

$$\mathbb{E}\left[\|\bar{g}(t)\|^2\right] \leq \frac{8L^2 d}{n} e_c(t-1) + \frac{32Ld}{3}\mathbb{E}[\delta(t-1)] + \frac{32Ld\Delta}{3} + u_t^2 L^2 d^2. \tag{29}$$

By plugging (29) into (13) and using the fact that f is μ-gradient dominated, we can get

$$\mathbb{E}[\delta(t)] \leq \left(1 - \frac{\eta_t\mu}{2}\right)\mathbb{E}[\delta(t-1)] + \frac{3\eta_t L^2}{2n} e_c(t-1) + \frac{16\eta_t^2 L^2 d\Delta}{3} + 2\eta_t u_t^2 L^2 d \tag{30}$$

where we have also used the fact that $\eta_t \leq 3\mu/(32L^2 d) \leq 1/(8Ld)$ under the conditions of Theorem 2. By combining this bound with Lemma 6, we get

$$\begin{bmatrix} e_c(t)/n \\ \mathbb{E}[\delta(t)] \end{bmatrix} \leq \begin{bmatrix} \frac{1+\rho^2}{2} & 4\rho^2 L\left(\frac{4}{3}\frac{d}{} + \frac{6\rho^2}{1-\rho^2}\right)\eta_t^2 \\ \frac{3L^2}{2}\eta_t & 1 - \frac{\eta_t\mu}{2} \end{bmatrix}\begin{bmatrix} e_c(t-1)/n \\ \mathbb{E}[\delta(t-1)] \end{bmatrix} + \upsilon_t \tag{31}$$

where

$$\upsilon_t = \begin{bmatrix} 4\eta_t^2\rho^2 L\Delta\left(\frac{4}{3}\frac{d}{} + \frac{6\rho^2}{1-\rho^2}\right) + \eta_t^2 u_t^2\rho^2 L^2\left(d^2 + \frac{6\rho^2}{1-\rho^2}\right) \\ \frac{16\eta_t^2 L^2\Delta d}{3} + 2\eta_t u_t^2 L^2 d \end{bmatrix}.$$

Since $\eta_t = 2\alpha_\eta/(t+t_0)$ and $u_t = O(1/\sqrt{t})$, straightforward calculation shows that

$$\left\| \begin{bmatrix} \frac{1+\rho^2}{2} & 4\rho^2 L\left(\frac{4}{3}\frac{d}{} + \frac{6\rho^2}{1-\rho^2}\right)\eta_t^2 \\ \frac{3L^2}{2}\eta_t & 1 - \frac{\eta_t\mu}{2} \end{bmatrix} \right\| = 1 - \frac{\alpha_\eta}{t} + O(1/t^2)$$

and $\|\upsilon_t\| = O(1/t^2)$. By (31) and the inequality

$$\prod_{s=t_1}^{t_2}\left(1 - \frac{\alpha_\eta}{2s}\right) \leq \left(\frac{t_1}{t_2+1}\right)^{\alpha_\eta/2} \tag{32}$$

for arbitrary $t_1 \leq t_2 + 1$, one can show that $\mathbb{E}[\delta(t)] = O(t^{-1/2})$.

Finally, by (28) and mathematical induction, one can show that

$$\frac{e_c(t)}{n} \leq \frac{16\alpha_\eta^2\rho^2 L}{\mu^2}\left(\frac{4}{3}\frac{d}{} + \frac{6\rho^2}{1-\rho^2}\right)\sum_{\tau=1}^{t}\frac{\mathbb{E}[\delta(\tau-1)]}{(\tau+t_0)^2}\left(\frac{1+\rho^2}{2}\right)^{t-\tau} + \frac{32\alpha_\eta^2\rho^2 L\Delta}{\mu^2(1-\rho^2)}\left(\frac{4}{3}\frac{d}{} + \frac{6\rho^2}{1-\rho^2}\right)\frac{1}{t^2} + o(t^{-2}).$$

Since $\mathbb{E}[\delta(t)] = O(t^{-1/2})$ and $\sum_{\tau=0}^{t-1}\lambda^\tau/(t-\tau)^\epsilon = O(t^{-\epsilon})$ for arbitrary $\epsilon > 0$, the first term on the right-hand side of the above inequality can be bounded by $o(t^{-2})$, which then leads to (18). By plugging (18) into (30) and using mathematical induction and (32), one can show (17).

APPENDIX C
PROOF SKETCH OF THEOREM 3

We further denote

$$s(t) := \begin{bmatrix} s^1(t)^\top & \cdots & s^n(t)^\top \end{bmatrix}^\top$$

and $e_c(t) := \|x(t) - 1_n \otimes \bar{x}(t)\|^2$, $e_g(t) := \|s(t) - 1_n \otimes \bar{g}(t)\|^2$. We recall that each f_i is assumed to be L-smooth, and note that $\frac{1}{n}\sum_{i=1}^n s^i(t) = \bar{g}(t)$.

We shall only provide a proof sketch of the bound (19). We first provide a lemma on the consensus procedure.

Lemma 8: Suppose $\eta L \le 1/6$. Then

$$\begin{bmatrix} \frac{5\eta}{2\sqrt{57}L} e_g(t) \\ e_c(t-1) \end{bmatrix} \le A \begin{bmatrix} \frac{5\eta}{2\sqrt{57}L} e_g(t-1) \\ e_c(t-2) \end{bmatrix} + v(t-2) \tag{33}$$

where

$$A := \begin{bmatrix} \frac{1+2\rho^2}{3} + \frac{18\rho^4(1+2\rho^2)}{1-\rho^2}\eta^2 L^2 & \frac{2\sqrt{57}\rho^2(1+2\rho^2)}{5(1-\rho^2)}\eta L \\ \frac{2\sqrt{57}\rho^2(1+2\rho^2)}{5(1-\rho^2)}\eta L & \frac{1+2\rho^2}{3} \end{bmatrix}$$

$$v(t) := \frac{2n\eta^3 L\rho^2(1+2\rho^2)}{3(1-\rho^2)} \begin{bmatrix} 2\|\nabla f(\bar{x}(t))\|^2 + \frac{5}{4}\frac{u_{t-1}^2 d}{\eta^2} \\ 0 \end{bmatrix}.$$

Proof Sketch: We first observe that

$$\begin{aligned} s(t) - 1_n \otimes \bar{g}(t) = {} & (W \otimes I_d)\,[s(t-1) - 1_n \otimes \bar{g}(t-1) \\ & + g(t) - g(t-1) - 1_n \otimes (\bar{g}(t) - \bar{g}(t-1))]\,. \end{aligned}$$

By Lemma 5, the Peter–Paul inequality and L-smoothness of f_i, it can be shown that

$$\begin{aligned} e_g(t) \le {} & \frac{1+2\rho^2}{3} e_g(t-1) \\ & + \frac{2\rho^2(1+2\rho^2)L^2}{1-\rho^2}\left(\|x(t-1) - x(t-2)\|^2 + n u_{t-1}^2 d\right). \end{aligned}$$

Then, by using $(W \otimes I_d)(1_n \otimes v) = 1_n \otimes v$ for any $v \in \mathbb{R}^d$, it can be shown that

$$\begin{aligned} & x(t-1) - x(t-2) \\ = {} & (W \otimes I_d - I_{nd})(x(t-2) - 1_n \otimes \bar{x}(t-2)) \\ & - \eta(W \otimes I_d)(s(t-1) - 1_n \otimes \bar{g}(t-1)) \\ & - \eta 1_n \otimes (\bar{g}(t-1) - \nabla f(\bar{x}(t-2))) - \eta 1_n \otimes \nabla f(\bar{x}(t-2)) \end{aligned}$$

and by using Lemma 5, the Peter–Paul inequality, L-smoothness of f_i, and that $\eta L \le 1/6$, it can be further shown that

$$\begin{aligned} \|x(t-1) - x(t-2)\|^2 \le {} & \frac{114}{25} e_c(t-2) + 9\eta^2\rho^2 e_g(t-1) \\ & 2\eta^2 n \|\nabla f(\bar{x}(t-2))\|^2 + \frac{1}{4} n u_{t-1}^2\, d. \end{aligned}$$

Therefore

$$\begin{aligned} e_g(t) \le {} & \left(\frac{1+2\rho^2}{3} + \frac{18\rho^4(1+2\rho^2)}{1-\rho^2}\eta^2 L^2\right) e_g(t-1) \\ & + \frac{228\rho^2(1+2\rho^2)}{25(1-\rho^2)} L^2 e_c(t-2) \\ & + \frac{2\rho^2(1+2\rho^2)}{1-\rho^2}\left(2\eta^2 L^2 n \|\nabla f(\bar{x}(t-2))\|^2 + \frac{5}{4} n L^2 u_{t-1}^2\, d\right). \end{aligned}$$

On the other hand, we have

$$\begin{aligned} e_c(t-1) = {} & \big\|(W \otimes I_d)[x(t-2) - 1_n \otimes \bar{x}(t-2) \\ & - \eta(s(t-1) - 1_n \otimes \bar{g}(t-1))]\big\|^2 \\ \le {} & \frac{1+2\rho^2}{3} e_c(t-2) + \frac{\rho^2(1+2\rho^2)}{1-\rho^2}\eta^2 e_g(t-1). \end{aligned}$$

We then get (33) by combining the last two inequalities. We make A symmetric so that it is more straightforward to compute its spectral norm. ■

Then, we have the following lemma on the evolution of $\delta(t)$.

Lemma 9: Suppose $\eta L \le 1/6$. Then

$$\begin{aligned} \delta(t) \le {} & \delta(t-1) - \frac{\eta}{3}\|\nabla f(\bar{x}(t-1))\|^2 \\ & + \frac{4\eta L^2}{3n} e_c(t-1) + \frac{\eta u_t^2\, L^2\, d}{3}. \end{aligned} \tag{34}$$

We are now ready to derive the results of Theorem 3. It can be shown that $\|A\| \le (2+\rho^2)/3$ when $\eta L \le \min\{1/6, (1-\rho^2)^2/(4\rho^2(3+4\rho^2))\}$. By taking the norms of both sides of (33) and using mathematical induction, it can be shown that

$$\begin{aligned} & \max\left\{\sum_{\tau=0}^{t-1} e_c(\tau), \frac{3\eta}{10\,L}\sum_{\tau=1}^{t} e_g(\tau)\right\} \\ \le {} & \frac{3nR_0}{1-\rho^2} + \frac{4n\eta^3 L\rho^2(1+2\rho^2)}{(1-\rho^2)^2}\sum_{\tau=0}^{t-2}\|\nabla f(\bar{x}(\tau))\|^2 \\ & + \frac{5n\eta L d\rho^2(1+2\rho^2)}{2(1-\rho^2)^2}\sum_{\tau=1}^{t-1} u_\tau^2. \end{aligned}$$

By plugging this bound into (34) and taking the telescoping sum, we get the bound (19).

APPENDIX D
PROOF SKETCH OF THEOREM 4

We shall only provide a proof sketch of the bound (23). We still denote

$$s(t) := \begin{bmatrix} s^1(t)^\top & \cdots & s^n(t)^\top \end{bmatrix}^\top$$

and $e_c(t) := \|x(t) - 1_n \otimes \bar{x}(t)\|^2$, $e_g(t) := \|s(t) - 1_n \otimes \bar{g}(t)\|^2$. In addition, we let $\theta := \mu/L$. By the L-smoothness of f, we have

$$f^* \le f(\bar{x}(t) - L^{-1}\nabla f(\bar{x}(t))) \le f(\bar{x}(t)) - \frac{1}{2\,L}\|\nabla f(\bar{x}(t))\|^2$$

which implies $\|\nabla f(\bar{x}(t))\|^2 \le 2L\delta(t)$. It can be shown by tedious calculation that when η is given by (22), we have $\|A\| \le 1 - (1-\rho^2)^2/21$. We also see that Lemma 8 still applies here as $\eta L \le 1/6$. By employing these observations and introducing

$$\chi := 1 - \frac{4}{49}\max_{\rho\in[0,1]} \rho^2(1+2\rho^2)(1-\rho^2)^3 \approx 0.9865$$

$$\sigma(t-1) := \frac{2\sqrt{2}L}{n\alpha\theta^{\frac{1}{3}}\sqrt{1-\chi}} \left\| \begin{bmatrix} \frac{5\eta}{2\sqrt{57}L} e_g(t) \\ e_c(t-1) \end{bmatrix} \right\|$$

we can show that

$$\sigma(t-1)\leq\left(1-\frac{(1-\rho^2)^2}{21}\right)\sigma(t-2)+\frac{\sqrt{2}\alpha\theta^{\frac{1}{3}}\sqrt{1-\chi}}{3}\eta L\delta(t-2)+\frac{5\sqrt{2}\rho^2(1+2\rho^2)(1-\rho^2)}{42\sqrt{1-\chi}}u_{t-1}^2\ Ld.$$

Then, by Lemma 9 and the assumption that f is μ-gradient dominated, we have

$$\begin{aligned}\delta(t-1)&\leq\left(1-\frac{2\eta\mu}{3}\right)\delta(t-2)+\frac{4\eta L^2}{3n}e_{\mathrm{c}}(t-2)+\frac{\eta L^2u_{t-1}^2\ d}{3}\\&\leq\left(1-\frac{2\eta\mu}{3}\right)\delta(t-2)+\frac{\sqrt{2}\alpha\theta^{\frac{1}{3}}\sqrt{1-\chi}}{3}\eta L\sigma(t-2)+\frac{\eta L^2u_{t-1}^2\ d}{3}.\end{aligned}$$

Therefore

$$\begin{bmatrix}\sigma(t-1)\\ \delta(t-1)\end{bmatrix}\leq B\begin{bmatrix}\sigma(t-2)\\ \delta(t-2)\end{bmatrix}+\begin{bmatrix}\frac{5\sqrt{2}\rho^2(1+2\rho^2)(1-\rho^2)}{14\sqrt{1-\chi}}\\ \eta L\end{bmatrix}\frac{u_{t-1}^2Ld}{3}\tag{35}$$

where

$$B:=\begin{bmatrix}1-\frac{1}{21}(1-\rho^2)^2 & \frac{1}{3}\sqrt{2(1-\chi)}\alpha\theta^{\frac{1}{3}}\eta L\\ \frac{1}{3}\sqrt{2(1-\chi)}\alpha\theta^{\frac{1}{3}}\eta L & 1-\frac{2}{3}\eta\mu\end{bmatrix}.$$

It can be shown that

$$\|B\|\leq 1-\frac{(1-\rho^2)^2}{25}\alpha\theta^{\frac{4}{3}}.$$

By plugging this bound into (35) and using mathematical induction, the bound (23) can be proved.

REFERENCES

[1] Y. Tang and N. Li, "Distributed zero-order algorithms for nonconvex multi-agent optimization," in *Proc. 57th Annu. Allerton Conf. Commun., Control, Comput.*, 2019, pp. 781–786.

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

[3] I.-A. Chen, "Fast distributed first-order methods," Master's thesis, Dept. Elect. Eng. Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, USA, 2012.

[4] J. Zeng and W. Yin, "On nonconvex decentralized gradient descent," *IEEE Trans. Signal Process.*, vol. 66, no. 11, pp. 2834–2848, Jun. 2018.

[5] S. Pu, A. Olshevsky, and I. Ch. Paschalidis, "A sharp estimate on the transient time of distributed stochastic gradient descent," 2019, *arXiv:1906.02702*.

[6] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu, "Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5336–5346.

[7] G. Scutari and Y. Sun, "Distributed nonconvex constrained optimization over time-varying digraphs," *Math. Program.*, vol. 176, nos. 1/2, pp. 497–544, 2019.

[8] A. Nedic, 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.

[9] Y. Nesterov and V. Spokoiny, "Random gradient-free minimization of convex functions," *Found. Comput. Math.*, vol. 17, no. 2, pp. 527–566, 2017.

[10] A. Nedić and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Trans. Autom. Control*, vol. 54, no. 1, pp. 48–61, Jan. 2009.

[11] 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.

[12] G. Qu and N. Li, "Harnessing smoothness to accelerate distributed optimization," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 3, pp. 1245–1260, Sep. 2018.

[13] G. Qu and N. Li, "Accelerated distributed Nesterov gradient descent," *IEEE Trans. Autom. Control*, vol. 65, no. 6, pp. 2566–2581, Jun. 2020.

[14] A. Nedić and A. Olshevsky, "Distributed optimization over time-varying directed graphs," *IEEE Trans. Autom. Control*, vol. 60, no. 3, pp. 601–615, Mar. 2015.

[15] J. Xu, S. Zhu, Y. C. Soh, and L. Xie, "Augmented distributed gradient methods for multi-agent optimization under uncoordinated constant stepsizes," in *Proc. 54th IEEE Conf. Decis. Control*, 2015, pp. 2055–2060.

[16] S. S. Ram, A. Nedić, and V. V. Veeravalli, "Distributed stochastic subgradient projection algorithms for convex optimization," *J. Optim. Theory Appl.*, vol. 147, no. 3, pp. 516–545, 2010.

[17] S. Omidshafiei, J. Pazis, C. Amato, J. P. How, and J. Vian, "Deep decentralized multi-task multi-agent reinforcement learning under partial observability," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, vol. 70, pp. 2681–2690.

[18] B. Charrow, N. Michael, and V. Kumar, "Cooperative multi-robot estimation and control for radio source localization," *Int. J. Robot. Res.*, vol. 33, no. 4, pp. 569–580, 2014.

[19] J. R. Marden, S. D. Ruben, and L. Y. Pao, "A model-free approach to wind farm control using game theoretic methods," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 4, pp. 1207–1214, Jul. 2013.

[20] P. Di Lorenzo and G. Scutari, "NEXT: In-network nonconvex optimization," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 2, no. 2, pp. 120–136, Jun. 2016.

[21] J. Kiefer *et al.*, "Stochastic estimation of the maximum of a regression function," *Ann. Math. Statist.*, vol. 23, no. 3, pp. 462–466, 1952.

[22] A. D. Flaxman, A. T. Kalai, and H. B. McMahan, "Online convex optimization in the bandit setting: Gradient descent without a gradient," in *Proc. 16th Annu. ACM-SIAM Symp. Discr. Algorithms*, 2005, pp. 385–394.

[23] F. Bach and V. Perchet, "Highly-smooth zero-th order online optimization," in *Proc. 29th Annu. Conf. Learn. Theory*, 2016, vol. 49, pp. 257–283.

[24] J. C. Duchi, M. I. Jordan, M. J. Wainwright, and A. Wibisono, "Optimal rates for zero-order convex optimization: The power of two function evaluations," *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2788–2806, May 2015.

[25] O. Shamir, "An optimal algorithm for bandit and zero-order convex optimization with two-point feedback," *J. Mach. Learn. Res.*, vol. 18, no. 52, pp. 1–11, 2017.

[26] 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. 57th IEEE Conf. Decis. Control*, 2018, pp. 4951–4958.

[27] Z. Yu, D. W. C. Ho, and D. Yuan, "Distributed randomized gradient-free mirror descent algorithm for constrained optimization," 2019, *arXiv:1903.04157*.

[28] B. T. Polyak, "Gradient methods for minimizing functionals," *USSR Comput. Math. Math. Phys.*, vol. 3, no. 4, pp. 864–878, 1963.

[29] S. Łojasiewicz, "A topological property of real analytic subsets," *Coll. du CNRS, Les équations aux dérivées partielles*, vol. 117, pp. 87–89, 1963.

[30] 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.

[31] M. Fazel, R. Ge, S. Kakade, and M. Mesbahi, "Global convergence of policy gradient methods for the linear quadratic regulator," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, vol. 80, pp. 1467–1476.

[32] O. Shamir, "Exponential convergence time of gradient descent for one-dimensional deep linear neural networks," in *Proc. 32nd Conf. Learn. Theory*, 2019, pp. 2691–2713.

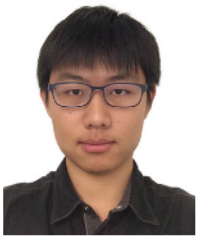
[33] A. H. Sayed, "Diffusion adaptation over networks," in *Academic Press Library in Signal Processing*, vol. 3. New York, NY, USA: Elsevier, 2014, pp. 323–453.

[34] Y. Tang, J. Zhang, and N. Li, "Distributed zero-order algorithms for nonconvex multi-agent optimization," 2019, *arXiv:1908.11444*.

[35] S. Ghadimi and G. Lan, "Stochastic first-and zeroth-order methods for nonconvex stochastic programming," *SIAM J. Optim.*, vol. 23, no. 4, pp. 2341–2368, 2013.

[36] A. Nedić and A. Olshevsky, "Stochastic gradient-push for strongly convex functions on time-varying directed graphs," *IEEE Trans. Autom. Control*, vol. 61, no. 12, pp. 3936–3947, Dec. 2016.

[37] A. Olshevsky, "Linear time average consensus and distributed optimization on fixed graphs," *SIAM J. Control Optim.*, vol. 55, no. 6, pp. 3990–4014, 2017.
[38] Z. Li, W. Shi, and M. Yan, "A decentralized proximal-gradient method with network independent step-sizes and separated convergence rates," *IEEE Trans. Signal Process.*, vol. 67, no. 17, pp. 4494–4506, Sep. 2019.
[39] S. Pu and A. Nedić, "A distributed stochastic gradient tracking method," in *Proc. 57th IEEE Conf. Decis. Control*, 2018, pp. 963–968.
[40] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proc. 4th Int. Symp. Inf. Process. Sensor Netw.*, 2005, pp. 63–70.



Yujie Tang (Member, IEEE) received the B.S. degree in electronic engineering from Tsinghua University, Beijing, China, in 2013, and the Ph.D. degree in electrical engineering from the California Institute of Technology, Pasadena, CA, USA, in 2019.

He is currently a Postdoctoral Fellow with the School of Engineering and Applied Sciences, Harvard University, Cambridge, MA, USA. His research interests include distributed and real-time/online optimization and their applications in cyber-physical networks.

Junshan Zhang (Fellow, IEEE) received the Ph.D. degree in electrical and computer engineering from Purdue University, West Lafayette, IN, USA, in 2000.

In 2000, he joined the School of Electrical, Computer, and Energy Engineering, Arizona State University, Tempe, AZ, USA, where he has been the Fulton Chair Professor since 2015. His research interests fall in the general field of information networks and data science, including communication networks, edge computing, and machine learning for Internet of Things, mobile social networks, and smart grid.

Dr. Zhang was the recipient of the ONR Young Investigator Award (2005), NSF CAREER Award (2003), IEEE Wireless Communication Technical Committee Recognition Award (2016) among others.

Na Li (Member, IEEE) received the B.S. degree in mathematics and applied mathematics from Zhejiang University, Hangzhou, China, in 2007, and the Ph.D. degree in control and dynamical systems from the California Institute of Technology, Pasadena, CA, USA, in 2013.

She is currently an Associate Professor with the School of Engineering and Applied Sciences, Harvard University, Cambridge, MA, USA. She was a Postdoctoral Associate with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology. Her research interests include design, analysis, optimization, and control of distributed network systems, with particular applications to cyber-physical network systems.

Dr. Li was the recipient of the NSF CAREER Award (2016), AFOSR Young Investigator Award (2017), ONR Young Investigator Award (2019), and Donald P. Eckman Award (2019), among others.