2024 IEEE 63rd Conference on Decision and Control (CDC)
December 16-19, 2024. MiCo, Milan, Italy

A Zeroth-Order Proximal Algorithm for Consensus Optimization

Chengan Wang, Zichong Ou and Jie Lu

Abstract— **This paper considers a consensus optimization problem, where all the nodes in a network, with access to the zeroth-order information of its local objective function only, attempt to cooperatively achieve a common minimizer of the sum of their local objectives. To address this problem, we develop ZoPro, a zeroth-order proximal algorithm, which incorporates a zeroth-order oracle for approximating Hessian and gradient into a recently proposed, high-performance distributed second-order proximal algorithm. We show that the proposed ZoPro algorithm, equipped with a dynamic stepsize, converges linearly to a neighborhood of the optimum in expectation, provided that each local objective function is strongly convex and smooth. Extensive simulations demonstrate that ZoPro converges faster than several state-of-the-art distributed zeroth-order algorithms and outperforms a few distributed second-order algorithms in terms of running time for reaching given accuracy.**

I. INTRODUCTION

This paper considers a widely studied distributed optimization problem, i.e., consensus optimization, where all the nodes in a network aim at reaching a consensus that minimizes the sum of their local cost functions. This problem arises in many real-world applications such as distributed machine learning [1] and resource allocation [2].

To date, a variety of distributed algorithms for convex consensus optimization have been proposed, in which each node only has access to certain information of its convex local cost function and can only communicate with its neighbors determined by the network topology. Most existing distributed optimization algorithms are first-order methods, which typically include primal (sub-)gradient methods [3]–[7] and dual/primal-dual (sub-)gradient methods [8]–[10]. These methods essentially require the nodes compute the (sub-)gradients of their primal or dual objectives.

The second-order methods, such as the Decentralized Broyden-Fletcher-Goldfarb-Shanno method (D-BFGS) [11], the Exact Second-Order Method (ESOM) [12] and the Decentralized Quadratically Approximated ADMM (DQM) [13], employ the objective Hessian matrices in addition to the objective gradients, potentially leading to faster convergence due to their more accurate approximations of some global objectives. Recently, Wu *et al*. have proposed a distributed second-order proximal algorithm called SoPro [14], which originates from the classic Method of Multipliers [15], while replacing the augmented Lagrangian function of the problem with its second-order approximation and introducing a separable quadratic proximal term to decouple the problem. SoPro achieves linear convergence under strong convexity and smoothness, and exhibits superior practical performance.

C. Wang, Z. Ou and J. Lu are with the School of Information Science and Technology, ShanghaiTech University, 201210, Shanghai, China. Email:{wangcha, ouzch, lujie}@shanghaitech.edu.cn.

All of the aforementioned algorithms are required to compute at least the first-order information (i.e., (sub-)gradients) and even the second-order information (i.e., Hessian matrices) of the problem. However, these pieces of information could be unavailable or too time-consuming to obtain in big data and large-scale network scenarios. Under such circumstances, zeroth-order algorithms [16]–[23] are effective approaches, whose updates only involve sampled function values instead of the exact (sub-)gradients and Hessian matrices. Various zeroth-order algorithms have been developed so far, which utilize zeroth-order estimators with different batch sizes to estimate the (sub-)gradients in some distributed first-order methods, including zeroth-order gradient tracking method [16], zeroth-order primal-dual methods [17], [18], zeroth-order decentralized (sub-)gradient descent [16], zeroth-order method with approximate projection [20], distributed zeroth-order projected gradient descent [21], distributed randomized zeroth-order mirror descent method [23], etc. To the best of our knowledge, zeroth-order Hessian estimators have barely been considered in distributed optimization. As distributed second-order algorithms often outperform the first-order ones in both accuracy and convergence rate, introducing zeroth-order oracles for Hessian estimation to distributed second-order methods is a promising direction.

In this paper, we propose a distributed zeroth-order proximal algorithm, referred to as ZoPro, for solving convex consensus optimization. ZoPro replaces the exact objective gradients and Hessian matrices in the distributed second-order algorithm SoPro [14] with their zeroth-order estimates, so that it significantly reduces the computational cost of SoPro and is applicable to the scenarios where the objective gradients and Hessian matrices are inaccessible or too costly to compute. ZoPro also inherits the appealing convergence performance of SoPro. It is shown to achieve a linear rate of convergence to a neighborhood of the optimal solution when the objective functions are strongly convex and smooth. Finally, the numerical experiments demonstrate that ZoPro outperforms a few state-of-the-art distributed zeroth-order methods in convergence speed and enjoys shorter running time than several well-noted second-order methods.

The rest of the paper is organized as follows: Section II describes the problem formulation, Section III develops the proposed ZoPro algorithm, Section IV provides convergence result, Section V presents the simulation results, and Section VI concludes the paper. Due to space limitation, we omit all the proofs, which can be found in [24]

Notations and definitions: For any differentiable function $f : \mathbb{R}^d \to \mathbb{R}$, $\nabla f(x)$ represents its gradient at $x \in \mathbb{R}^d$ and

if f is twice-differentiable, $\nabla^2 f(x)$ represents its Hessian matrix. $\mathbf{O}_d$ and $\mathbf{I}_d$ represent the $d \times d$ zero matrix and identity matrix, respectively, and $\mathbf{0}_d$ and $\mathbf{1}_d$ represent the d-dimensional all-zero and all-one vectors, respectively. Define $\mathbb{Z}^+$ and $\mathbb{N}$ as the sets of positive integers and non-negative integers, respectively. Also, $\otimes$ denotes the Kronecker product, $\|\cdot\|$ denotes the $\mathcal{L}_2$ norm and $\langle\cdot,\cdot\rangle$ denote the inner product. Besides, $\text{diag}(A_1,\ldots,A_n)$ represents the block diagonal matrix consisting of the diagonal blocks $A_1,\ldots,A_n$. $[P]_{ij}$ denotes the (i,j)-entry of matrix P. Given $A = A^T \in \mathbb{R}^{d\times d}$ and $\mathbf{x} \in \mathbb{R}^d$, $\|\mathbf{x}\|_A^2 = \mathbf{x}^T A\mathbf{x}$. $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$ represent A's smallest and largest eigenvalues, respectively. $A^\dagger$ denotes A's pseudo-inverse and $A^\perp$ represents the orthogonal complement of A. $v \sim \mathcal{N}(\mu,\Sigma)$ represents a Gaussian random vector v with mean μ and covariance matrix Σ. A function $f : \mathbb{R}^d \to \mathbb{R}$ is μ-strongly convex if f is differentiable and

$$f(y) \geq f(x) + \nabla f(x)^T(y-x) + \frac{\mu}{2}\|y-x\|^2 \ \forall x,y \in \mathbb{R}^d$$

for some $\mu > 0$. f is L-smooth if f is differentiable and

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x-y\|, \ \forall x,y \in \mathbb{R}^d$$

for some $L > 0$. Finally, the directional derivative of f at point x along direction d is denoted by $f'(x;d) = \lim_{\alpha\to 0}\frac{f(x+\alpha d)-f(x)}{\alpha}$.

II. PROBLEM FORMULATION

We consider solving

$$\underset{x\in\mathbb{R}^d}{\text{minimize}} \sum_{i\in\mathcal{V}} f_i(x) \tag{1}$$

over a network modeled as a connected and undirected graph $\mathcal{G} = (\mathcal{V},\mathcal{E})$, where $\mathcal{V} = \{1,2,\ldots,N\}$ is the set of nodes and $\mathcal{E} \subseteq \{\{i,j\} \subseteq \mathcal{V}\times\mathcal{V} \mid i \neq j\}$ is the set of the bidirectional links. For each $i \in \mathcal{V}$, we denote the set of its neighbors by $\mathcal{N}_i = \{j \in \mathcal{V} \mid \{i,j\} \in \mathcal{E}\}$. Here, $f_i : \mathbb{R}^d \to \mathbb{R}$ is the local cost/objective function associated with node $i \in \mathcal{V}$ and each node i only communicates with its neighbors in $\mathcal{N}_i$.

Let $x_i \in \mathbb{R}^d$ be node i's local copy of the global optimization variable x and $\mathbf{x}$ be the concatenation of all the x_i's, i.e. $\mathbf{x} = (x_1^T,\ldots,x_N^T)^T \in \mathbb{R}^{Nd}$. Let $P = P^T$ be a weight matrix corresponding to the network $\mathcal{G}$ given by

$$[P]_{ij} = \begin{cases} \sum_{s\in\mathcal{N}_i} p_{is}, & i = j, \\ -p_{ij}, & j \in \mathcal{N}_i, \\ 0, & \text{otherwise}, \end{cases} \qquad \forall i,j \in \mathcal{V},$$

where $p_{ij} = p_{ji} > 0 \ \forall \{i,j\} \in \mathcal{E}$.

Due to the fact that $\mathcal{G}$ is connected, the null space of P is span$\{\mathbf{1}_N\}$, so that we can rewrite problem (1) as

$$\begin{aligned} \underset{x\in\mathbb{R}^{Nd}}{\text{minimize}} \ & f(\mathbf{x}) = \sum_{i\in\mathcal{V}} f_i(x_i) \\ \text{subject to } & W^{\frac{1}{2}}\mathbf{x} = \mathbf{0}_{Nd}, \end{aligned} \tag{2}$$

where $W = P \otimes \mathbf{I}_d \succeq \mathbf{O}_{Nd}$ and the equality constraint means that $x_1,\ldots,x_n$ are identical [14].

We impose the following assumption on problem (1).

Assumption 1: Each f_i is m_i-strongly convex, twice continuously differentiable and M_i-smooth, where m_i, $M_i > 0$.

Note that Assumption 1 guarantees the uniqueness of the optimal solution x^* to problem (1).

III. ALGORITHM DEVELOPMENT

In this section, we develop a distributed zeroth-order algorithm for solving problem (2).

A. SoPro Algorithm

We first quickly review the second-order proximal (SoPro) algorithm proposed in [14].

SoPro solves (2) in a primal-dual fashion as follows:

$$\mathbf{x}^{k+1} = \mathbf{x}^k - (\nabla^2 f(\mathbf{x}^k) + D)^{-1}(\nabla f(\mathbf{x}^k) + \rho W\mathbf{x}^k + \mathbf{q}^k), \tag{3}$$

$$\mathbf{q}^{k+1} = \mathbf{q}^k + \rho W\mathbf{x}^{k+1}, \tag{4}$$

with the initialization $\mathbf{q}^0 = \mathbf{0}_{Nd}$, where $\mathbf{x}^k$ is the global primal variable and $\mathbf{q}^k = W^{\frac{1}{2}}\mathbf{v}^k$ is a change of variable with $\mathbf{v}^k$ being the dual variable associated with the constraint in (2). The primal update (3) intends to minimize an augmented-Lagrangian-like function constructed in the following way: The augmented Lagrangian function of (2) with penalty $\frac{\rho}{2}\|W^{\frac{1}{2}}\mathbf{x}\|^2$, $\rho > 0$ is first replaced by its second-order approximation at $\mathbf{x}^k$ to reduce computational cost. Then, a separable quadratic proximal term $\frac{1}{2}(\mathbf{x}-\mathbf{x}^k)^T(\nabla^2 f(\mathbf{x}^k) + D)(\mathbf{x}-\mathbf{x}^k)$ is employed as a substitute for the non-separable term in the above approximate augmented Lagrangian function to enable fully distributed implementation, where $D = \text{diag}(D_1,\ldots,D_N)$ is a symmetric block diagonal matrix with each $D_i \in \mathbb{R}^{d\times d}$ satisfying $\nabla^2 f(\mathbf{x}^k) + D \succ \mathbf{O}_{Nd}$. The dual update (4) emulates dual gradient ascent.

Observe that the updates (3) and (4) of SoPro require calculating accurate first-order and second-order information of the objective function f, which could be a tough challenge when handling big data and large-scale problems. The high computational complexity of SoPro motivates the development of zeroth-order oracles for efficiently estimating the gradients and Hessian in SoPro.

B. Zeroth-order Oracle

Next, we provide a zeroth-order oracle for estimating the gradients and Hessian matrices in SoPro's updates (3) and (4). To do so, consider the following smoothed approximation of the objective function f:

$$f_\mu(\mathbf{x}) \triangleq \frac{1}{(2\pi)^{Nd/2}}\int_{\mathbb{R}^{Nd}} f(\mathbf{x}+\mu\mathbf{u})e^{-\frac{\|\mathbf{u}\|^2}{2}}d\mathbf{u}, \tag{5}$$

where $\mathbf{u} \sim (0,\mathbf{I}_{Nd}) \in \mathbb{R}^{Nd}$ is a Gaussian random vector and $\mu > 0$ is a parameter to control the smoothness level [25, Section 2]. Note that the smoothed approximation f_μ is guaranteed to be differentiable. We will show more properties of f_μ in Section IV. Let $\widetilde{g}_\mu$ and $\widetilde{H}_\mu$ represent the gradient and Hessian estimation of the above Gaussian smoothing function f_μ, which are defined according to [22] as follows:

$$\widetilde{g}_\mu(\mathbf{x}) = \frac{1}{b}\sum_{j=1}^{b}\frac{f(\mathbf{x}+\mu u_j) - f(\mathbf{x})}{\mu}u_j. \tag{6}$$

$$\widetilde{H}_\mu(\mathbf{x}) = \text{diag}\left(\widetilde{H}_{\mu,1}(x_1),\ldots,\widetilde{H}_{\mu,N}(x_N)\right), \tag{7}$$

$$\widetilde{H}_{\mu,i}(x_i)$$
$$=\frac{1}{b}\sum_{j=1}^{b}\frac{f_i(x_i+\mu u_j)+f_i(x_i-\mu u_j)-2f_i(x_i)}{2\mu^2}u_j u_j^T,$$
$$\forall i=1,\ldots,N,$$

where $b \in \mathbb{Z}^+$ is the batch size, $u_j \sim \mathcal{N}(0,\mathbf{I}_d) \in \mathbb{R}^d$, $j=1,\ldots,b$ are Gaussian random vectors. The zeroth-order oracle (6)–(7) for gradient and Hessian matrix estimation only needs to sample $(2b+1)N$ points from the local objective functions, which is much less costly than computing the exact gradient and Hessian matrix. It can be verified that the zeroth-order gradient estimation (6) is an unbiased estimator for ∇f_μ [19], i.e., $\mathbf{E}\left[\widetilde{g}_\mu(\mathbf{x})\right]=\nabla f_\mu(\mathbf{x})$.

C. ZoPro Algorithm

In this subsection, we incorporate the zeroth-order oracle (6)–(7) into SoPro, yielding a zeroth-order proximal algorithm, referred to as ZoPro.

We first replace $\nabla f(\mathbf{x}^k)$ and $\nabla^2 f(\mathbf{x}^k)$ in (3) and (4) with $\widetilde{g}_\mu(\mathbf{x}^k)$ and $\widetilde{H}_\mu(\mathbf{x}^k)$, which gives

$$\mathbf{x}^{k+1}=\mathbf{x}^k-(\widetilde{H}_\mu(\mathbf{x}^k)+D)^{-1}(\widetilde{g}_\mu(\mathbf{x}^k)+\rho W\mathbf{x}^k+\mathbf{q}^k), \tag{8}$$

$$\mathbf{q}^{k+1}=\mathbf{q}^k+\rho W\mathbf{x}^{k+1}, \tag{9}$$

where, similar to SoPro, $D=\mathrm{diag}(D_1,\ldots,D_N)\in\mathbb{R}^{Nd\times Nd}$ is a symmetric block diagonal matrix such that $\widetilde{H}_\mu(\mathbf{x})+D\succ\mathbf{O}_{Nd}\ \forall\mathbf{x}\in\mathbb{R}^{Nd}$, or equivalently, $\widetilde{H}_{\mu,i}(x_i)+D_i\succ\mathbf{0}_d\ \forall x\in\mathbb{R}^d\ \forall i=1,\ldots,N$. The starting point $\mathbf{q}^0$ is set to $\mathbf{q}^0\in S^\perp$, where $S=\left\{\mathbf{x}\in\mathbb{R}^{Nd}|x_1=\cdots=x_N\right\}$ and $S^\perp=\left\{\mathbf{x}\in\mathbb{R}^{Nd}|x_1+\cdots+x_N=\mathbf{0}_d\right\}$, so that $\mathbf{q}^k\in S^\perp\ \forall k\geq 0$ due to (9). For simplicity, we set $\mathbf{q}^0=\mathbf{0}_{Nd}$.

Since $\widetilde{g}_\mu(\mathbf{x}^k)$ and $\widetilde{H}_\mu(\mathbf{x}^k)$ are only estimated values of $\nabla f(\mathbf{x}^k)$ and $\nabla^2 f(\mathbf{x}^k)$, (8) and (9) may not converge to the exact optimum like SoPro. To overcome this issue, we introduce a backtracking line search strategy with a dynamic stepsize to bound the sequence $\left\{\mathbf{x}^k\right\}$. We set the search direction to $\mathbf{d}^k=-(\widetilde{H}_\mu(\mathbf{x}^k)+D)^{-1}(\widetilde{g}_\mu(\mathbf{x}^k)+\rho W\mathbf{x}^k+\mathbf{q}^k)$, and then modify (8) to

$$\mathbf{x}^{k+1}=\mathbf{x}^k+A^k\mathbf{d}^k. \tag{10}$$

Here, $A^k=\mathrm{diag}(\alpha_1^k,\alpha_2^k,\ldots,\alpha_N^k)\otimes\mathbf{I}_d\in\mathbb{R}^{Nd\times Nd}$ and $\alpha_i^k,\ i\in\mathcal{V}$ is the local stepsize of node i determined by the Armijo condition [26, Eq. (1)], i.e. $f_i(x_i^k+\alpha_i^k d_i^k)\leq f(x_i^k)+c\alpha_i^k f_i'(x_i^k;d_i^k)$, where $c>0$ is the stepsize control parameter, $f_i'(x_i^k;d_i^k)$ is the directional derivative of f_i at x_i^k along node i's local search direction d_i^k and d_i^k is the i-th d-dimensional block of $\mathbf{d}^k$. In light of the Armijo condition, it is always possible to find a sufficiently small stepsize $\alpha_i^k>0$ that satisfies the condition. In practice, by choosing an appropriate parameter c, the backtracking line search may require several iterations. Nevertheless, this approach ensures that our algorithm remains competitive in numerical experiments as is shown in Section V.

The primal update (10) and the dual update (9) with initialization $\mathbf{q}^0=\mathbf{0}_{Nd}$ constitute a zeroth-order proximal algorithm, referred to as ZoPro, whose distributed implementation is described in Algorithm 1.

Algorithm 1 Zeroth-Order Proximal Algorithm (ZoPro)

1: **Initialization:**
All the nodes agree on the batch size $b\in\mathbb{Z}^+$, the smoothness parameter $\mu>0$, the penalty parameter $\rho>0$, and the stepsize control parameter $c\in(0,1)$. Generate b random vectors $u_j\sim\mathcal{N}(0,\mathbf{I}_d)\ \forall j=1,\ldots,b$.
2: Each node $i\in\mathcal{V}$ chooses D_i such that $\widetilde{H}_{\mu,i}(x)+D_i\succ\mathbf{O}_d\ \forall x\in\mathbb{R}^d$ and sets the initial stepsize $\alpha_i^0=1$.
3: Every pair of neighboring nodes $\{i,j\}\in\mathcal{E}$ set $p_{ij}=p_{ji}$ to some positive value.
4: Each node $i\in\mathcal{V}$ sets $x_i^0\in\mathbb{R}^d$ arbitrarily and $q_i^0=\mathbf{0}_d$. Then, it sends x_i^0 to every neighbor $j\in\mathcal{N}_i$.
5: Upon receiving $x_j^0\ \forall j\in\mathcal{N}_i$, each node $i\in\mathcal{V}$ sets $y_i^0=\sum_{j\in\mathcal{N}_i}p_{ij}(x_i^0-x_j^0)$.
6: **for** $k\geq 0$ **do**
7: Each node $i\in\mathcal{V}$ computes Hessian estimate $\widetilde{H}_{\mu,i}(x_i^k)=b^{-1}\sum_{j=1}^{b}\frac{f_i(x_i^k+\mu u_j)+f_i(x_i^k-\mu u_j)-2f_i(x_i^k)}{2\mu^2}u_j u_j^T$.
8: Each node $i\in\mathcal{V}$ computes gradient estimate $\widetilde{g}_{\mu,i}(x_i^k)=\frac{1}{b}\sum_{j=1}^{b}\frac{f_i(x_i^k+\mu u_j)-f_i(x_i^k)}{\mu}u_j$.
9: Each node $i\in\mathcal{V}$ computes the search direction $d_i^k=-(\widetilde{H}_{\mu,i}(x_i^k)+D_i)^{-1}(\widetilde{g}_{\mu,i}(x_i^k)+\rho y_i^k+q_i^k)$.
10: Each node $i\in\mathcal{V}$ determines the stepsize α_i^k such that $f_i(x_i^k+\alpha_i^k d_i^k)\leq f(x_i^k)+c\alpha_i^k f_i'(x_i^k;d_i^k)$.
11: Each node $i\in\mathcal{V}$ updates $x_i^{k+1}=x_i^k+\alpha_i^k d_i^k$ and sends x_i^{k+1} to every neighbor $j\in\mathcal{N}_i$.
12: Upon receiving $x_j^{k+1}\ \forall j\in\mathcal{N}_i$, each node $i\in\mathcal{V}$ updates $y_i^{k+1}=\sum_{j\in\mathcal{N}_i}p_{ij}(x_i^{k+1}-x_j^{k+1})$ and $q_i^{k+1}=q_i^k+\rho y_i^{k+1}$.
13: **end for**

In Algorithm 1, each node i maintains a local primal variable $x_i^k\in\mathbb{R}^d$ and a local dual variable $q_i^k\in\mathbb{R}^d$, which are the i-th d-dimensional block of $\mathbf{x}^k$ and $\mathbf{q}^k$. Also, we let it maintain an auxiliary variable $y_i^k\in\mathbb{R}^d$ such that $\mathbf{y}^k=\left((y_1^k)^T,\ldots,(y_N^k)^T\right)^T=W\mathbf{x}^k$ for better presentation.

The existing zeroth-order distributed optimization methods such as distributed zeroth-order gradient tracking method [16], distributed zeroth-order primal-dual method [17], [18], distributed zeroth-order projected gradient descent [21] and distributed randomized zeroth-order mirror descent method [23] all use zeroth-order information to approximate the objective gradients only. In contrast, ZoPro includes zeroth-order estimates for both gradients and Hessian matrices. This may accelerate the convergence as ZoPro adopts potentially more accurate approximations of the global objective than other zeroth-order methods.

IV. CONVERGENCE ANALYSIS

This section provides the convergence analysis of ZoPro. First, we analyze some properties of f_i and f_i'.

Proposition 1: Let $f_i:\mathbb{R}^d\to\mathbb{R}$ be a L-smooth function and let $\left\{x_i^k\right\}$ be the sequence generated by $x_i^{k+1}=x_i^k+\alpha_i^k d_i^k$, where α_i^k is the stepsize determined by the backtracking line search and d_i^k is the corresponding search direction. Denote the directional derivative of f_i as f_i'. Then

one of the following statements is true:
(i) $f_i(x_i^k) \to -\infty$ as $k \to \infty$.
(ii) The sequence $\{\|d_i^k\|\}$ diverges.
(iii) For every infinite subsequence $J \subseteq \mathbb{N}$ for which $\{d_i^k : k \in J\}$ is bounded, we have

$$\lim_{k \in J, k \to \infty} f_i'(x_i^k; d_i^k) = 0.$$

From Proposition 1, we have $\lim_{k\to\infty} f_i'(x_i^k; d_i^k) = 0$ for f_i, $i \in \mathcal{V}$, which indicates that either the gradient $\nabla f_i(x_i^k)$ and the search direction d_i^k are orthogonal as $k \to \infty$ or $\nabla f_i(x_i^k)$ equals 0 as $k \to \infty$, and both results can terminate the backtracking line search. From [27, Theorem 3.2], backtracking line search method guarantees convergence of the generated sequence $\{\mathbf{x}^k\}$. By continuous mapping theorem, $\{\nabla f(\mathbf{x}^k)\}$ is convergent, and thus $\{\|\nabla f(\mathbf{x}^k)\|\}$ is bounded. For simplicity, we denote the upper bound as K.

Assumption 1 implies f is strongly convex for some $m \in (0, \min_i m_i]$ and smooth for some $M \geq \max_i M_i$. [19, Eq. (9)] derives a bound of the difference between ∇f_μ and $\widetilde{g}_\mu$.

$$\mathbf{E}\left[\|\widetilde{g}_\mu(\mathbf{x}) - \nabla f_\mu(\mathbf{x})\|^2\right] \leq \frac{2Nd\left(\mu^2 M^2 Nd + K^2\right)}{b} \triangleq G_1^2. \tag{11}$$

Besides, the difference between ∇f_μ and ∇f is also bounded in [25, Eq. (28)] as follows:

$$\|\nabla f_\mu(\mathbf{x}) - \nabla f(\mathbf{x})\|^2 \leq \frac{\mu^2}{4} M^2 (Nd+3)^3 \triangleq G_2^2. \tag{12}$$

Moreover, let $f_{\mu,i}$ be the smoothed approximation of function f_i, i.e. $f_{\mu,i}(x_i) = \frac{1}{(2\pi)^{d/2}} \int_{\mathbb{R}^d} f(\mathbf{x} + \mu u_i) e^{-\frac{\|u_i\|^2}{2}} du_i$, where, similarly, $u_i \sim (0, \mathbf{I}_d) \in \mathbb{R}^d$, $i = 1, \ldots, N$ is Gaussian random vectors. According to [25, Section 2], the smoothing function $f_{\mu,i}$ can preserve all characteristics of f_i. For example, $f_{\mu,i}$ is guaranteed to be m_i-strongly convex and M_i-smooth if f_i is m_i-strongly convex and M_i-smooth. $f_{\mu,i}$ is twice continuously differentiable if f_i is twice continuously differentiable. Assumption 1 also implies $m_i \mathbf{I}_d \preceq \nabla^2 f_i(x) \preceq M_i \mathbf{I}_d$, $i = 1, \ldots, N$ $\forall x \in \mathbb{R}^d$. Let $\Lambda_m = \text{diag}(m_1, m_2, \ldots, m_N) \otimes \mathbf{I}_d \succ \mathbf{O}_{Nd}$ and $\Lambda_M = \text{diag}(M_1, M_2, \ldots, M_N) \otimes \mathbf{I}_d \succ \mathbf{O}_{Nd}$. Besides, in order to provide a bound of the stepsize in backtracking line search, we define the smallest stepsize in the whole process as $\underline{\alpha}$, i.e. $\underline{\alpha} = \min_{i,k} \alpha_i^k > 0$ for $i = 1, \ldots, N$ and $k = 0, 1, \ldots$. We have $\underline{\alpha} \in (0, 1]$ since $\alpha_i^0 = 1$. Let $R = \underline{\alpha}^{-1}(\frac{\Lambda_M + \Lambda_m}{2} + D) \in \mathbb{R}^{Nd \times Nd}$ and $Q = \text{diag}(\rho R, \mathbf{I}_{Nd})$. For simplicity, define $H^k = \widetilde{H}_\mu(\mathbf{x}^k) + D$.

Similar to [22, Eq. (3.1)], we impose another assumption to bound the difference between $\widetilde{H}_{\mu,i}(x_i)$ and $\nabla^2 f_i(x_i)$.

Assumption 2: The estimated Hessian $\widetilde{H}_{\mu,i}(x_i)$ satisfies

$$\theta \widetilde{H}_{\mu,i}(x_i) \preceq \nabla^2 f_i(x_i) \preceq (2-\theta)\widetilde{H}_{\mu,i}(x_i),$$

for $i = 1, \ldots, N$ and some $\theta \in (0, 1]$.

Parameter θ measures how accurate $\widetilde{H}_{\mu,i}(x_i)$ approximates $\nabla^2 f_i(x_i)$. Specifically, $\widetilde{H}_{\mu,i}(x_i)$ reduces to the exact Hessian $\nabla^2 f_i(x_i)$ when $\theta = 1$. The way of constructing zeroth-order estimate for Hessian (7) may satisfy Assumption 2 with proper parameter values such as sufficiently large b, small μ and evenly distributed u_j, as is stated in [22]. Practically, we tend to select a small batch size b, which is much smaller than dimension d, to reduce computational complexity.

From Assumption 1 and 2, we have

$$\frac{1}{2-\theta}\Lambda_m \preceq \frac{1}{2-\theta}\nabla^2 f(\mathbf{x}) \preceq \widetilde{H}_\mu(\mathbf{x}) \preceq \frac{1}{\theta}\nabla^2 f(\mathbf{x}) \preceq \frac{1}{\theta}\Lambda_M. \tag{13}$$

Let $\bar{\Lambda} = \underline{\alpha}^{-1}\left(\frac{1}{\theta}\Lambda_M - \frac{\Lambda_M + \Lambda_m}{2}\right)$. The convergence analysis relies on the following condition

$$D \succ \frac{\Lambda_M}{2\eta} + \rho(W + \mathbf{I}_{Nd}) + \left(\frac{2}{\theta} - \frac{3}{2}\right)\Lambda_M - \frac{3}{2}\Lambda_m + \left(\frac{1}{\theta}\Lambda_M - \frac{\Lambda_M + \Lambda_m}{2}\right)^2, \tag{14}$$

for any $\eta > 1$. With (14), it is guaranteed that $\widetilde{H}_\mu(\mathbf{x}) + D \succ \mathbf{O}_{Nd}$ $\forall \mathbf{x} \in \mathbb{R}^{Nd}$ since $\widetilde{H}_\mu(\mathbf{x}) \succeq \frac{1}{2-\theta}\Lambda_m$.

For better presentation, let $\mathbf{z}^k = ((\mathbf{x}^k)^T, (\mathbf{v}^k)^T)^T$ and $\mathbf{z}^* = ((\mathbf{x}^*)^T, (\mathbf{v}^*)^T)^T$. Also, let $\lambda_W > 0$ be the smallest nonzero eigenvalue of W. The main convergence result of ZoPro is provided below.

Theorem 1: Suppose Assumptions 1 and 2 hold. Assume (14) holds for some $\eta > 1$. Then, for any $\beta > \underline{\alpha}^{-1}$ and $\gamma > \frac{2m(\eta-1)+\eta+\beta}{\eta-1}$, $\mathbf{z}^k$ converges linearly to a neighborhood of $\mathbf{z}^*$ in expectation, i.e. there exists $\delta \in (0, 1)$ such that for each $k \geq 0$,

$$\mathbf{E}\left[\left\|\mathbf{z}^{k+1} - \mathbf{z}^*\right\|_Q^2\right] \leq (1-\delta)\mathbf{E}\left[\left\|\mathbf{z}^k - \mathbf{z}^*\right\|_Q^2\right] + G, \tag{15}$$

$$\limsup_{k\to\infty} \mathbf{E}\left[\left\|\mathbf{z}^k - \mathbf{z}^*\right\|_Q^2\right] \leq \frac{G}{\delta}. \tag{16}$$

In particular, given any $c_1, c_2 > 0$, $G = \rho(\eta + \frac{1-\eta}{\gamma})G_2^2 + 2(G_1^2 + G_2^2) + \frac{2\delta(1+c_1)(G_1+G_2)^2}{\lambda_W}$ and

$$\delta = \sup_{c_1,c_2>0} \min\left\{\frac{\rho\lambda_W \kappa_{\beta,\eta}}{2\underline{\alpha}^{-2}(1+c_1)\left\|\frac{1}{\theta}\Lambda_M + D\right\|^2}, \frac{1}{(1+1/c_1)(1+c_2)}, \frac{\delta_c}{\lambda_{\max}(\mathcal{B}/\rho)}\right\}, \tag{17}$$

in which $\mathcal{B} = \frac{(1+1/c_1)(1+1/c_2)\Lambda_M^2}{\lambda_W} + \rho R$, $\delta_c = (2m-\gamma)(1-\eta) - \eta - \beta$ and $\kappa_{\beta,\eta} = \lambda_{\min}(R - \frac{\Lambda_M}{2\eta} - \frac{\bar{\Lambda}^2}{\beta} - 2\bar{\Lambda} - \rho(\mathbf{I}_{Nd} + W)) > 0$.

Subsequently, we discuss the influence of the objective function and the network topology on the convergence rate of ZoPro. From (15), note that δ mainly depends on M, m and λ_W. To see this, let f_i $\forall$ $i \in \mathcal{V}$ be identically strongly convex with parameter m and smooth with parameter M such that $0 < m < M$. Let $W = \mathbf{I}_{Nd} - A \otimes \mathbf{I}_d$, where $A = A^T$ is a doubly stochastic matrix. It can be shown that larger m, smaller M and larger λ_W (which suggests denser connectivity of $\mathcal{G}$) lead to larger δ and a faster convergence speed of ZoPro.

Also, we discuss the factors that affect the ultimate optimality error, i.e., the expected distance between $\mathbf{z}^k$ and $\mathbf{z}^*$ as $k \to \infty$. From (16), this expected distance mainly depends on μ, b and M. It can be shown that smaller μ (accurate smoothed approximation), larger b (enough sample points for zeroth-order oracle) and smaller M (well-conditioned

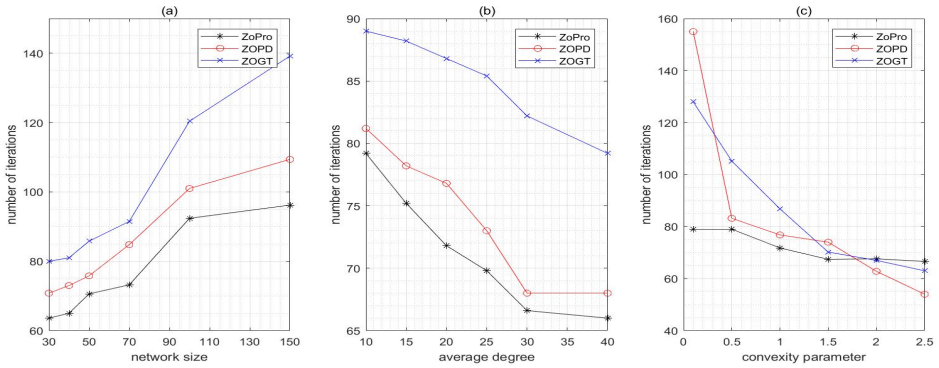

Fig. 1: Convergence performance of ZoPro, ZOPD and ZOGT

objective function) contribute to a smaller expected error. However, we need to control b within a moderate range to avoid high computational cost in constructing Hessian and gradient estimates.

V. NUMERICAL EXPERIMENTS

This section illustrates the practical convergence performance of ZoPro and its comparisons with related algorithms.

In the numerical experiment, we consider the following logistic regression problem with $\mathcal{L}_2$ regularization: All the nodes need to cooperatively minimize the objective function $f(x) = \sum_{i\in\mathcal{V}}\left(\frac{\lambda}{2N}\|x_i\|^2 + \sum_{l=1}^{q_i}\log\left(1+\exp(-v_{il}\mathbf{u}_{il}^T x_i)\right)\right)$, where $x_i \in \mathbb{R}^d$, λ is a regularized parameter, N is the number of nodes, q_i is the sample number assigned to node i, $\mathbf{u}_{ij}$, $j = 1,\ldots,q_i$ is the data and $v_{ij} \in \{-1,+1\}$, $j = 1,\ldots,q_i$ is the corresponding label. Here, we set $q_i = 5\ \forall i \in \mathcal{V}$.

A. Comparison with Zeroth-order Methods

We compare the convergence performance of ZoPro with some other existing zeroth-order optimization methods, including zeroth-order gradient-tracking method (ZOGT) [16] and zeroth-order primal-dual method (ZOPD) [17].

We set $d = 20$. In order to test how the problem and network characteristics influence the convergence performance, we consider three parameters, i.e., the network size N, the average node degree $d_a = \sum_{i\in\mathcal{V}}|\mathcal{N}_i|/N$ and the convexity parameter λ. Accordingly, we run three groups of experiments, each of which fixes two of these parameters and varies the other. We express each experiment as a triplet (N, d_a, λ), and set the three experiment groups to (G1) $(N, 20, 1)$, $N = 30, 40, 50, 70, 100, 150$, (G2) $(50, d_a, 1)$, $d_a = 10, 15, 20, 25, 30, 40$ and (G3) $(50, 20, \lambda)$, $\lambda = 0.1, 0.5, 1, 1.5, 2, 2.5$. For each value of (N, d_a, λ), we generate 10 random scenarios and plot their average in our figures. In each scenario, the undirected network is randomly generated with the given N and d_a.

The algorithm parameters of ZoPro are selected moderately. We let the smoothness parameter $\mu = 0.05$, the batch size $b = 50$ and the stepsize control parameter $c = 0.1$. We terminate the algorithms when the average error of all the nodes $\sum_{i\in\mathcal{V}}\|x_i^k - x^*\|^2/N$ drops below 10^{-4} and remains there for 100 more iterations. Therefore, we define the number of iterations needed for convergence as $\min\left\{k : \sum_{i\in\mathcal{V}}\|x_i^{k+t} - x^*\|^2/N \le 10^{-4},\ 0 \le t \le 100\right\}$.

Figure 1 plots the number of iterations needed for convergence of ZoPro, ZOPD and ZOGT with (N, d_a, λ) given by (G1), (G2) and (G3), respectively. Observe that smaller and denser networks as well as larger convexity parameters essentially lead to faster convergence for ZoPro, ZOGT and ZOPD. Compared to ZOGT and ZOPD, our proposed ZoPro requires the fewest iterations to reach the convergence criterion in most of the cases.

B. Comparison with Second-order Methods

To illustrate the computational efficiency of ZoPro, we make a comparison of ZoPro and the well-noted second-order methods ESOM [12], DQM [13] and SoPro [14] in terms of running time. Here, we consider the same problem form as before, and set (N, d_a, λ, d) to be $(30, 10, 1, 20)$ and $(100, 20, 1, 30)$ to simulate a medium-scale problem and a large-scale problem.

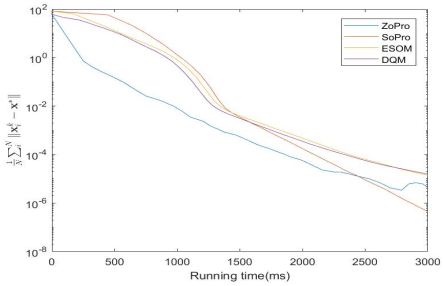

Fig. 2: Convergence performance of ZoPro, ESOM, DQM and SoPro for the medium-scale problem

In Figure 2 for the medium-scale problem, ZoPro consistently takes shorter running time for reaching the same accuracy than ESOM and DQM. Compared to SoPro which ZoPro originates from, ZoPro requires less time to reach the accuracy 10^{-4}. However, when the accuracy is enhanced to 10^{-6}, SoPro is a better option. In Figure 3 for the large-scale problem, the advantage of ZoPro against all the other three second-order methods becomes more prominent and ZoPro takes the shortest time in achieving any accuracy above 10^{-6}.

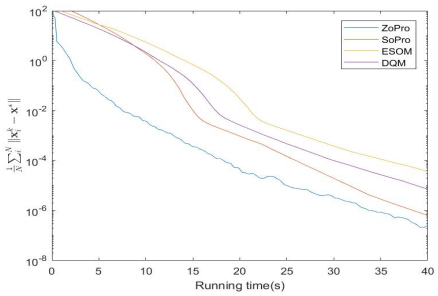

Fig. 3: Convergence performance of ZoPro, ESOM, DQM and SoPro for the large-scale problem

VI. CONCLUSION

We develop a zeroth-order proximal algorithm (ZoPro) for solving consensus optimization problems over undirected networks. ZoPro approximates exact gradients and Hessian matrices in a powerful second-order method SoPro using a zeroth-order oracle, which significantly reduces the computational complexity, particularly in solving large-scale problems. ZoPro inherits some appealing features of SoPro, including full decentralization and fast convergence. We show that ZoPro achieves linear convergence to a neighborhood of the optimum in expectation when the problem is strongly convex and smooth. We also demonstrate the fast convergence and computational efficiency of ZoPro by comparing it with several state-of-the-art distributed zeroth-order and second-order algorithms through extensive simulations.

REFERENCES

[1] E. P. Xing, Q. Ho, P. Xie, and D. Wei, "Strategies and principles of distributed machine learning on big data," *Engineering*, vol. 2, no. 2, pp. 179-195, 2016.

[2] A. Beck, A. Nedić, A. Ozdaglar, and M. Teboulle, "An O(1/k) gradient method for network resource allocation problems," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 1, pp. 64–73, 2014.

[3] W. Shi, Q. Ling, G. Wu, and W. Yin, "EXTRA: an exact first-order algorithm for decentralized consensus optimization," *SIAM Journal on Optimization*, vol. 25, no. 2, pp. 944–966, 2015.

[4] W. Shi, Q. Ling, G. Wu, and W. Yin, "A proximal gradient algorithm for decentralized composite optimization," *IEEE Transactions on Signal Processing*, vol. 63, no. 22, pp. 6013–6023, 2015.

[5] J. Lu and C. Tang, "Zero-gradient-sum algorithms for distributed convex optimization: The continuous-time case," *IEEE Transactions on Automatic Control*, vol. 57, no. 9, pp. 2348–2354, 2012.

[6] C. Xi and U. Khan, "DEXTRA: a fast algorithm for optimization over directed graphs," *IEEE Transactions on Automatic Control*, vol. 62, no. 10, pp. 4980–4993, 2017.

[7] C. Xi, R. Xin, and U. Khan, "ADD-OPT: accelerated distributed directed optimization," *IEEE Transactions on Automatic Control*, vol. 63, no. 5, pp. 1329–1339, 2018

[8] P. Bianchi, W. Hachem, and F. Iutzeler, "A coordinate descent primal dual algorithm and application to distributed asynchronous optimization," *IEEE Transactions on Automatic Control*, vol. 61, no. 10, pp. 2947–2957, 2016.

[9] A. Makhdoumi and A. Ozdaglar, "Convergence rate of distributed ADMM over networks," *IEEE Transactions on Automatic Control*, vol. 62, no. 10, pp. 5082–5095, 2017.

[10] T. Chang, A. Nedić, and A. Scaglione, "Distributed constrained optimization by consensus-based primal-dual perturbation method," *IEEE Transactions on Automatic Control*, vol. 59, no. 6, pp. 1524–1538, 2014.

[11] M. Eisen, A. Mokhtari, and A. Ribeiro, "Decentralized quasi-newton methods," *IEEE Transactions on Signal Processing*, vol. 65, no. 10, pp. 2613–2628, 2017.

[12] A. Mokhtari, W. Shi, Q. Ling, and A. Ribeiro, "A decentralized second-order method with exact linear convergence rate for consensus optimization," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 4, pp. 507–522, 2016.

[13] A. Mokhtari, W. Shi, Q. Ling, and A. Ribeiro, "DQM: decentralized quadratically approximated alternating direction method of multipliers," *IEEE Transactions on Signal Processing*, vol. 64, no. 19, pp. 5158–5173, 2016.

[14] X. Wu, Z. Qu, and J. Lu, "A second-order proximal algorithm for consensus optimization," *IEEE Transactions on Automatic Control*, vol. 66, no. 4, pp. 1864–1871, 2020.

[15] A. Agarwal, O. Dekel, and L. Xiao, "Optimal algorithms for online convex optimization with multi-point bandit feedback," *Conference on Learning Theory*, pp. 28–40, 2010.

[16] Y. Tang, J. Zhang, and N. Li, "Distributed zero-order algorithms for nonconvex multiagent optimization," *IEEE Transactions on Control of Network Systems*, vol. 8, no. 1, pp. 269-281, 2020.

[17] X. Yi, S. Zhang, T. Yang, T. Chai, and K.J ohansson, "Linear convergence of first-and zeroth-order primal-dual algorithms for distributed nonconvex optimization," *IEEE Transactions on Automatic Control*, vol. 67, no. 8, pp. 4194-4201, 2021.

[18] D. Hajinezhad and M. Zavlanos, "Gradient-free multi-agent nonconvex nonsmooth optimization," *IEEE Conference on Decision and Control*, pp. 4939-4944, 2018.

[19] D. Hajinezhad, M. Hong, and A. Garcia, "ZONE: Zeroth-order nonconvex multiagent optimization over networks," *IEEE Transactions on Automatic Control*, vol. 64, no. 10, pp. 3995-4010, 2019.

[20] D. Yuan, D. Ho, and S. Xu, "Zeroth-order method for distributed optimization with approximate projections," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 2, pp. 284–294, 2015.

[21] Y. Pang and G. Hu, "Randomized gradient-free distributed optimization methods for a multiagent system with unknown cost function," *IEEE Transactions on Automatic Control*, vol. 65, no. 1, pp. 333-340, 2019.

[22] H. Ye, Z. Huang, C. Fang, C. Li, and T. Zhang, "Hessian-aware zeroth-order optimization for black-box adversarial attack," 2018, *arXiv:1812.11377*.

[23] Z. Yu, D. Ho, and D. Yuan, "Distributed randomized gradient-free mirror descent algorithm for constrained optimization," *IEEE Transactions on Automatic Control*, vol. 67, no. 2, pp. 957-964, 2021.

[24] C. Wang, Z. Ou, J. Lu, "A Zeroth-Order Proximal Algorithm for Consensus Optimization," 2024, *arXiv:2406.09816*.

[25] Y. Nesterov and V. Spokoiny, "Random gradient-free Minimization of convex functions," *Foundations of Computational Mathematics*, vol. 17, no. 2, pp. 527-566, 2017.

[26] L. Armijo, "Minimization of functions having Lipschitz continuous first partial derivatives," *Pacific Journal of Mathematics*, vol. 16, no. 1, pp. 1-3, 1966.

[27] P. A. Absil, R. Mahony, and B. Andrews, "Convergence of the iterates of descent methods for analytic cost functions," *SIAM Journal on Optimization*, vol. 16, no. 2, pp. 531-547, 2005.