IEEE Control Systems Letters paper presented at
2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

ZO-JADE: Zeroth-order Curvature-Aware Distributed Multi-Agent Convex Optimization

Alessio Maritan, *Graduate Student Member, IEEE*, and Luca Schenato, *Fellow, IEEE*

***Abstract*— In this work we address the problem of convex optimization in a multi-agent setting where the objective is to minimize the average of local cost functions whose derivatives are not available (e.g. black-box models). Moreover agents can only communicate with local neighbors according to a connected network topology. Zeroth-order (ZO) optimization has recently gained increasing attention in federated learning and multi-agent scenarios exploiting finite-difference approximations of the gradient using from 2 (directional gradient) to $2d$ (central difference full gradient) evaluations of the cost functions, where d is the dimension of the problem. The contribution of this work is to extend ZO distributed optimization by estimating the curvature of the local cost functions via finite-difference approximations. In particular, we propose a novel algorithm named ZO-JADE, that by adding just one extra point, i.e. $2d+1$ in total, allows to simultaneously estimate the gradient and the diagonal of the local Hessian, which are then combined via average tracking consensus to obtain an approximated Jacobi descent. Guarantees of semi-global exponential stability are established via separation of time-scales. Extensive numerical experiments on real-world data confirm the efficiency and superiority of our algorithm with respect to several other distributed zeroth-order methods available in the literature based on only gradient estimates.**

I. Introduction

Optimization is unquestionably the main pillar and enabler of several real-world applications, as it allows to get the best solution for any given problem, improve the performance of processes and efficiently implement modern technologies. However, most of the optimization techniques heavily rely on the knowledge of the derivatives of the objective function, which may be expensive or infeasible to obtain [1]. Such situations occur when the derivative is not available in analytical closed form, when automatic differentiation cannot be applied, in simulation-based optimization and when dealing with black-box models. Addressing these scenarios, the zeroth-order (ZO) and derivative-free optimization (DFO) branches have appeared in the literature, aiming to remove the dependency from the derivative information. Notable applications include adversarial attacks and explanations for deep neural networks, hyper-parameters tuning and policy search in reinforcement learning [2].

The most basic and widely used approach to ZO optimization is to replace the true gradient with an estimator based on the Kiefer–Wolfowitz scheme [3], which approximates the derivative by finite-differences of function values along a set of search directions. The DFO branch, instead, includes direct-search methods, interpolation models and trust-region frameworks [4].

A common assumption in the ZO field is that evaluating the objective function is a costly process, for example involving complex simulations, and indeed the convergence rate is often expressed in terms of number of function calls. For this reason, in this work we extract as much information as possible from the query points, estimating the curvature information of the objective function and exploiting the latter to speed up convergence. We note that in the ZO literature most of the works limit themselves to extracting the gradient information, and only few of them take into account also the second-order derivative. Among them, [5] uses a forward-difference gradient estimator in which the search directions are random Gaussian vectors scaled by the square root of the inverse Hessian matrix, and the latter is estimated using either a power-based algorithm or some heuristic methods. A similar idea is used in [6], where two perturbations with inverse Hessian covariance are used to estimate both the gradient and the Hessian by central differencing. Other examples of second-order methods based on finite-difference approximations are [7], which applies Newton's method along a single direction, and [8], which proposes a ZO version of the well known limited memory BFGS. Finally, [9] employs the second-order Stein's identity to estimate the Hessian and presents a cubic-regularized Newton method. Regarding the DFO field, there are many algorithms involving quadratic models constructed by means of interpolation, Least Squares, Lagrange Polynomials [10] or minimum Frobenius norm fitting [11].

However, all the aforementioned works consider only the centralized setting. In many relevant cases, such as distributed machine learning and estimation, robotic swarms and wireless sensor networks, it is required to spread the optimization process across a set of agents, which are often endowed with limited communication capabilities. Notably, the multi-agent setup does not require all the data and the processing power to be owned by a single centralized authority, increasing the scalability and the privacy guarantees of the algorithm. Currently, all the existing second-order solutions based on ZO estimates do not enjoy these appealing features. To address this shortcoming we propose a novel distributed second-order method for ZO optimization, which to the best of our knowledge is the first of its kind. Our algorithm, named ZO-JADE, takes advantage of the second-order derivative to obtain a Jacobi-like descent direction,

This work was supported in part by the Italian PNRR under Grant DM 352-2022; in part by Maschio Gaspardo S.p.A.; and in part by the Italian MUR Project "Realtime Control of 5G Wireless Networks" under Grant PRIN-2017NS9FEY.

The authors are with the Department of Information Engineering, University of Padova, Italy alessio.maritan@phd.unipd.it; schenato@dei.unipd.it

i.e. the gradient is re-scaled by the inverse of the diagonal of the Hessian. Using tools from Lyapunov theory and singular perturbation theory we formally prove the semi-global exponentially fast convergence of ZO-JADE to a point arbitrarily close to the minimum. The theoretical results are validated by numerical experiments, namely distributed ridge regression and one-vs-all classification on public datasets. These tests show that our algorithm, compared to several other zeroth-order multi-agent methods, converges much faster and reaches a possibly better solution.

Notation: We denote with I_d the d-dimensional identity matrix, with e_k the k-th canonical vector, and with $\mathbb{0}$ and $\mathbb{1}$ the vectors whose components are all equal to 0 and 1, respectively. We indicate with $\odot$ and $\oslash$ the element-wise Hadamard product and division. The Euclidean or spectral norm is $\|\cdot\|$ and coincides with the largest singular value of the argument. We use the operator $[\cdot]_i$ to select the i-th element of the argument. The third-order derivative tensor of $f : \mathbb{R}^d \to \mathbb{R}$ evaluated at x is denoted by $\left(D^3 f\right)(x)$.

II. Problem Formulation

In this paper we address the fully distributed optimization scenario, where n agents can communicate with each other through bi-directional links according to a connected network topology. Each agent i has access to a local cost function $f_i : \mathbb{R}^d \to \mathbb{R}$, and the goal of the network is to cooperatively seek the optimal solution of the unconstrained minimization problem

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

We consider the setting in which only zeroth-order information is available, namely it is only possible to evaluate the value of the local functions at the desired points, without access to gradients or higher-order derivatives. To provide a rigorous analysis we characterize the objective function with the following assumption.

Assumption 1 (Strong Convexity and Smoothness): Let the global cost f be three times continuously differentiable with Lipschitz continuous derivatives and m-strongly convex, i.e. there exist positive constants L_1, L_2, L_3 and m such that

$$\begin{aligned} m I_d \leq \nabla^2 f(x) \leq L_1 I_d \quad & \forall x \in \mathbb{R}^d, \\ \left\|\nabla^2 f(x) - \nabla^2 f(y)\right\| \leq L_2 \|x - y\| \quad & \forall x, y \in \mathbb{R}^d, \\ \left\|\left(D^3 f\right)(x) - \left(D^3 f\right)(y)\right\| \leq L_3 \|x - y\| \quad & \forall x, y \in \mathbb{R}^d. \end{aligned}$$

We are interested in consensus-based algorithms, in which all the agents converge to the same optimal value of the variable to be optimized by just communicating with their neighbors. To do so, each agent updates its local variables using a weighted average of its neighbors' information, where the weights are provided by a mixing matrix P. The assumption below provides a formal definition of the distributed scenario under consideration.

Assumption 2 (Communication network): Let the communication network be represented by the time-invariant graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ whose vertices $\mathcal{N} = \{1, \ldots, n\}$ and edges $\mathcal{E}$ are the set of agents and the available communication links, respectively. Assume that the graph is undirected and connected, and that each node can only communicate with its single-hop neighbors. Let the network be associated with the consensus matrix $P \in \mathbb{R}^{n \times n}$, whose generic entry p_{ij} is positive if edge $(i, j) \in \mathcal{E}$ and zero otherwise. Choose P to be symmetric and doubly stochastic, i.e. such that $P\mathbb{1} = \mathbb{1}$ and $\mathbb{1}^T P = \mathbb{1}^T$. A matrix that satisfies these requirements can be constructed without knowledge of the entire network topology using e.g. the Metropolis-Hastings weights [12].

III. Zeroth-order Oracles

Our algorithm requires to evaluate the first and second derivative of the local cost functions at the current point. Since in the zeroth-order setting these information are not available, we estimate them using central finite-difference schemes along orthogonal directions. Given a generic function f and a small positive scalar μ we adopt the following approximations of the gradient and of the Hessian's diagonal:

$$\begin{aligned} \hat{\nabla} f(\mu, x) &:= \sum_{k=1}^{d} \frac{f(x + \mu e_k) - f(x - \mu e_k)}{2\mu} e_k, \\ \hat{\nabla}^2 f(\mu, x) &:= \sum_{k=1}^{d} \frac{f(x + \mu e_k) - 2f(x) + f(x - \mu e_k)}{\mu^2} e_k. \end{aligned} \tag{2}$$

We emphasize that many algorithms, such as [13] and [14], require $2d$ function queries only for the gradient estimation. In this work instead, by querying the function also at the current iterate, i.e. $f(x)$, which is in any case necessary to evaluate the progress, we estimate also the diagonal of the Hessian almost for free.

Remark 1: Obtaining these estimates involves $2d+1$ function queries, which can be computationally demanding for high-dimensional problems. Alternatively, one could resort to randomized schemes in which directional derivatives are estimated along a small number of random vectors. However, using this type of estimators increases the total communication cost and does not guarantee that fewer total function evaluations are needed for convergence. As also argued in [13], which considers both a 2-points randomized gradient estimator and a $2d$-points deterministic one, there exists a trade-off between the convergence rate and the ability to handle high-dimensional problems. In the interests of space, in this work we focus only on deterministic estimators, considering the use of stochastic estimators as a possible research avenue.

Using Taylor expansions it can be shown that in case of noiseless function evaluations the approximation errors of the estimators are bounded by

$$\left\|\hat{\nabla} f(\mu, x) - \nabla f(x)\right\| \leq \frac{\sqrt{d} L_2 \mu^2}{6},$$

$$\left|\left[\hat{\nabla}^2 f(\mu, x)\right]_i - \left[\nabla^2 f(x)\right]_{ii}\right| \leq \frac{L_3 \mu^2}{12} \quad \forall i = 1, \ldots, d,$$

and that the Jacobian of the gradient estimator takes the form

$$\frac{\partial \hat{\nabla} f(x)}{\partial x} = \nabla^2 f(x) + R(\mu, x), \quad \left|[R(\mu, x)]_{ij}\right| \leq \frac{L_3 \mu^2}{6},$$

$\forall i, j, x$. Additionally, the estimators are Lipschitz continuous:

$$\left\|\hat{\nabla} f(\mu, x) - \hat{\nabla} f(\mu, y)\right\| \leq \left(L_1 + \frac{\mu\sqrt{d}L_2}{2}\right) \|x - y\|,$$

$$\left\|\hat{\nabla}^2 f(\mu, x) - \hat{\nabla}^2 f(\mu, y)\right\| \leq \left(L_2\sqrt{d} + \frac{\mu\sqrt{d}L_3}{3}\right) \|x - y\|.$$

We refer the interested reader to the technical report [15] for detailed derivations.

Remark 2: The approximation errors of the estimators and the absolute values of the elements of the matrix $R(\mu, x)$ are bounded by functions of μ^2. This is due to the symmetry of central-difference scheme, and shows that the estimates are very reliable.

Since for a quadratic function $L_2 = L_3 = 0$, we have the following result:

Lemma 1: In case of quadratic functions the gradient estimator and the Hessian estimator are exact, i.e. they coincide with the true gradient and with the diagonal of the true Hessian matrix, respectively.

Let $x^\star$ be the unique global minimizer of $f(x)$. The following Lemma, whose proof can be found in [15], states that for any value of μ there exists a unique point $\Gamma(\mu) \in \mathbb{R}^d$ for which the gradient estimator is the zero vector. Moreover, the distance between $\Gamma(\mu)$ and the optimal solution is bounded by a term proportional to μ^2.

Lemma 2: Let $f(x)$ be a function which satisfies Assumption 1 and $\hat{\nabla} f(\mu, x)$ be defined by (2). Then there exist $\bar{\mu}, \delta > 0$ and a continuously differentiable function $\Gamma : \mathbb{R} \to \mathbb{R}^d$ such that for all $\mu \leq \bar{\mu}$ there is a unique $x = \Gamma(\mu) \in \mathbb{R}^d$ such that $\|\Gamma(\mu) - x^\star\| \leq \delta\mu^2$ and $\hat{\nabla} f(\mu, \Gamma(\mu)) = 0$.
To lighten the notation, from now on we will denote the gradient and Hessian estimators as $\hat{\nabla} f(x)$ and $\hat{\nabla}^2 f(x)$, keeping the dependence from the parameter μ implicit.

IV. Proposed Algorithm

In this section we present ZO-JADE to efficiently solve the distributed convex optimization problem (1). Our method can be seen as the ZO counterpart of a special case of [16], with the additional difference that consensus is performed also directly on the variable to be optimized. In particular, we employ a Newton-type update which only requires the diagonal of the Hessian, sometimes referred to as Jacobi descent [17], from which the name ZO-JADE (JAcobi DEcentralized). This allows to benefit from the accelerated convergence rate which is typical of second-order methods, while just estimating the curvature along the canonical basis. Also, since the Hessian estimate is a vector, the storage requirement and the communication overhead are linear in d, and computing the inverse of the Hessian is straightforward.

We now provide an intuitive explanation of the algorithm. The idea is to approximate each local $f_i(x)$ with a d-dimensional parabola whose axes are aligned with the canonical basis, e.g.

Algorithm ZO-JADE

Initialize:
  Arbitrary $x_i(0) \in \mathbb{R}^d$, $\epsilon > 0$, mixing matrix P.
  $y_i(0) = g_i(0) = \mathbb{0}$,
  $z_i(0) = h_i(0) = \mathbb{0} \quad \forall i \in \mathcal{N}$.
for $t = 1, 2, \dots$ **do**
  for all $i \in \mathcal{N}$ **do**
    Compute $\hat{\nabla} f_i(x_i(t-1))$, $\hat{\nabla}^2 f_i(x_i(t-1))$ using (2).
    $g_i(t) = \hat{\nabla}^2 f_i(x_i(t-1)) \odot x_i(t-1) - \hat{\nabla} f_i(x_i(t-1))$,
    $h_i(t) = \hat{\nabla}^2 f_i(x(t-1))$,
    $y_i(t) = \sum_{j=1}^n p_{ij}[y_j(t-1) + g_j(t) - g_j(t-1)]$,
    $z_i(t) = \sum_{j=1}^n p_{ij}[z_j(t-1) + h_j(t) - h_j(t-1)]$,
    $x_i(t) = (1-\epsilon)\sum_{j=1}^n p_{ij} x_j(t-1) + \epsilon y_i(t) \oslash z_i(t)$.
  end for
end for

$$f_i(x) \approx \hat{f}_i(x) := \frac{1}{2} x^T (a_i \odot x) + b_i^T x + c_i,$$

$$n\hat{f}(x) = \sum_{i=1}^n \hat{f}_i(x) = \frac{1}{2} x^T \left(\sum_{i=1}^n a_i \odot x\right) + \sum_{i=1}^n b_i^T x + \sum_{i=1}^n c_i,$$

for suitable $a_i \in \mathbb{R}^d$ with positive entries, $b_i \in \mathbb{R}^d$, $c_i \in \mathbb{R}$. At the current iterate $\{x_i\}_{i=1}^n$, we match the derivatives of the model with the estimates provided by the ZO oracles:

$$\nabla \hat{f}_i(x_i) = a_i \odot x_i + b_i = \hat{\nabla} f_i(x_i),$$
$$\nabla^2 \hat{f}_i(x_i) = a_i = \hat{\nabla}^2 f_i(x_i).$$

We then want to move towards the minimum of the approximate global function, namely

$$\operatorname*{argmin}_x \hat{f}(x) = -\left(\frac{1}{n}\sum_{i=1}^n b_i\right) \oslash \left(\frac{1}{n}\sum_{i=1}^n a_i\right) \tag{3}$$
$$= \left(\sum_{i=1}^n \hat{\nabla}^2 f_i(x_i) \odot x_i - \hat{\nabla} f_i(x_i)\right) \oslash \left(\sum_{i=1}^n \hat{\nabla}^2 f_i(x_i)\right).$$

To do so, we define the local variables $\in \mathbb{R}^d$

$$g_i(t) = \hat{\nabla}^2 f_i(x_i(t-1)) \odot x_i(t-1) - \hat{\nabla} f_i(x_i(t-1)),$$
$$h_i(t) = \hat{\nabla}^2 f_i(x_i(t-1)), \tag{4}$$

and we estimate the means in (3) using gradient and Hessian average tracking. As the name suggests, this technique introduces two additional consensus variables which track the average of the derivatives over the network:

$$y_i(t) = \sum_{j=1}^n p_{ij}(y_j(t-1) + g_j(t) - g_j(t-1)),$$
$$z_i(t) = \sum_{j=1}^n p_{ij}(z_j(t-1) + h_j(t) - h_j(t-1)). \tag{5}$$

Indeed, using recursion it is easy to verify that by initializing $y_i(0) = g_i(0),\ z_i(0) = h_i(0)\ \forall i$ it holds $\forall t > 0$

$$\sum_{i=1}^{n} y_i(t) = \sum_{i=1}^{n} g_i(t), \qquad \sum_{i=1}^{n} z_i(t) = \sum_{i=1}^{n} h_i(t). \quad (6)$$

Assuming now that the mean of $\{x_i\}_{i=1}^n$ is almost constant, the local variables $\{y_i\}_{i=1}^n$ and $\{z_i\}_{i=1}^n$ will eventually reach consensus, becoming equal to the numerator and denominator of (3). The almost stationarity of the mean of $\{x_i\}_{i=1}^n$ is satisfied by adopting the time-scales separation framework. According to the latter, a sub-system can be regarded at steady state provided that it evolves sufficiently fast compared to the rest of the system. Equivalently, a fast system can consider a slow one as a fixed constraint. In our case, we need the consensus (5) to be part of the fast dynamics, while the dynamics

$$\sum_{i=1}^{n} x_i(t) = \sum_{i=1}^{n} \left[(1-\epsilon) \sum_{j=1}^{n} p_{ij} x_j(t-1) + \epsilon y_i(t) \oslash z_i(t) \right]$$

must be sufficiently slow. We will prove that there exists a positive scalar $\bar{\epsilon}$ such that for any $\epsilon \le \bar{\epsilon}$ the time-scales separation is verified and the algorithm converges exponentially fast. Moreover, property (6) guarantees that we are moving towards the minimum (3) of the global objective, allowing the convex combination coefficient ϵ to be constant.

Numerical simulations show that the consensus on the current iterates in the update rule of $x_i(t)$, which is not present in [16], brings significant performance improvements to the algorithm, at the price of an additional communication variable. However, the main concern in the zeroth-order setting is to minimize the number of function queries, which in our case is directly proportional to the number of iterations. Accordingly, the cost of the communication overhead becomes negligible when compared to the gain in terms of convergence rate.

V. Convergence analysis

In this section we prove the semi-global exponential convergence of ZO-JADE to a solution arbitrarily close to the true minimum of the optimization problem (1). Compared to [16], both the additional consensus step on x and the presence of inexact derivative estimators pose significant challenges and require careful treatment. In particular, the former makes the algorithm incompatible with the time-scales separation principle in [18], which was needed in the convergence proof. For this reason, in this work we extend that principle and provide the following more general result.

Theorem 1 (Separation of time-scales): The origin of the autonomous discrete-time system

$$\begin{cases} \bar{x}(k) = \bar{x}(k-1) + \epsilon \phi\left(\bar{x}(k-1), \xi(k-1)\right) \\ \xi(k) = \varphi\left(\xi(k-1), \bar{x}(k-1)\right) + \epsilon \Psi\left(\xi(k-1), \bar{x}(k-1)\right) \end{cases}$$

is semi-globally exponentially stable for a suitable choice of $\epsilon > 0$, provided that the following conditions are verified:

(i) The functions φ, Ψ, ϕ are locally uniformly Lipschitz.

(ii) There exists $\xi^\star$ such that $\xi^\star(\bar{x}) = \varphi\left(\xi^\star(\bar{x}), \bar{x}\right)\ \forall \bar{x}$.

(iii) At the origin, $\phi\left(\mathbb{0}, \xi^*(\mathbb{0})\right) = \mathbb{0}$ and $\Psi\left(\xi^*(\mathbb{0}), \mathbb{0}\right) = \mathbb{0}$.

(iv) There exists a twice differentiable function $V(\bar{x})$ and positive constants c_1, c_2, c_3, c_4 such that

$$c_1 \|\bar{x}\|^2 \le V(\bar{x}) \le c_2 \|\bar{x}\|^2,$$
$$\frac{\partial V}{\partial \bar{x}} \phi\left(\bar{x}, \xi^*(\bar{x})\right) \le -c_3 \|\bar{x}\|^2, \qquad \left\| \frac{\partial V}{\partial \bar{x}} \right\| \le c_4 \|\bar{x}\|.$$

Proof: In the interests of space, we only provide a sketch of the proof. We define $\xi'(k) := \xi(k) - \xi^\star(\bar{x}(k))$ and a Lyapunov function $W(\bar{x}, \xi'(k))$ for the fast dynamics $\xi'(k)$. We consider $U = V + W$ and show that the quantity $U(\bar{x}(k+1), \xi'(k+1)) - U(\bar{x}(k), \xi'(k))$ is always negative. The full proof is contained in Appendix A of the technical report [15]. ■

Below we present our convergence result, which provides theoretical guarantees on the stability and performance of the algorithm.

Theorem 2 (Convergence of algorithm ZO-JADE): Under Assumption 1, there exists a positive scalar $\bar{\epsilon}$ such that for any $\epsilon \in [0,\ \bar{\epsilon}]$ ZO-JADE is guaranteed to converge semi-globally and exponentially fast to $\Gamma(\mu)$.

Proof: The proof is based on a reformulation of our algorithm which leads to a discrete-time system compatible with Theorem 1. First, we rewrite the iterations of ZO-JADE to obtain an equivalent autonomous system with the same structure as the one considered by Theorem 1. In the second part of the proof we demonstrate that the system satisfies all the assumptions of Theorem 1, and invoking the latter we get the desired result. To simplify the notation let us stack the local variables and define $x(t) = [x_1(t), \dots, x_n(t)]^T \in \mathbb{R}^{n \times d}$. Similarly, define $g(t), h(t), y(t), z(t), \hat{\nabla} f(x(t)), \hat{\nabla}^2 f(x(t)) \in \mathbb{R}^{n \times d}$. We introduce the auxiliary variables

$$\bar{x}(t) = \frac{1}{n} \mathbb{1}\mathbb{1}^T x(t), \qquad \tilde{x}(t) = x(t) - \bar{x}(t),$$

which are respectively the mean of $x(t)$ and the displacement from it. This allows us to rewrite the algorithm as a dynamical system and to decouple the fast dynamics

$$\begin{aligned}
&g(t) = \varphi_g = \hat{\nabla}^2 f(x(t-1)) \odot x(t-1) - \hat{\nabla} f(x(t-1)), \\
&h(t) = \varphi_h = \hat{\nabla}^2 f(x(t-1)), \\
&y(t) = \varphi_y = P\left(y(t-1) + g(t) - g(t-1)\right), \\
&z(t) = \varphi_z = P\left(z(t-1) + h(t) - h(t-1)\right), \\
&\tilde{x}(t) = \varphi_{\tilde{x}} + \epsilon \Psi_{\tilde{x}} = \left(P - \frac{1}{n}\mathbb{1}\mathbb{1}^T\right) \tilde{x}(t-1) \\
&\quad + \epsilon \left[\left(I_n - \frac{1}{n}\mathbb{1}\mathbb{1}^T\right) y(t) \oslash z(t) - \left(P - \frac{1}{n}\mathbb{1}\mathbb{1}^T\right) \tilde{x}(t-1) \right]
\end{aligned}$$

from the slow dynamics

$$\begin{aligned}
\bar{x}(t) &= \bar{x}(t-1) + \epsilon \left(\frac{1}{n} \mathbb{1}\mathbb{1}^T [y(t) \oslash z(t)] - \bar{x}(t-1) \right) \\
&= \bar{x}(t-1) + \epsilon \phi_{\bar{x}}.
\end{aligned}$$

Since $x(t) = \bar{x}(t) + \tilde{x}(t)$ and the values $g(t),\ h(t),\ y(t),\ z(t)$ ultimately depend only on quantities at time $t-1$, the system is autonomous. By defining $\xi \coloneqq \{g, h, y, z, \tilde{x}\}$ and

$$\varphi\left(\xi(k-1), \bar{x}(k-1)\right) \coloneqq \varphi_g + \varphi_h + \varphi_y + \varphi_z + \varphi_{\tilde{x}},$$
$$\Psi\left(\xi(k-1), \bar{x}(k-1)\right) \coloneqq \Psi_{\tilde{x}},\ \phi\left(\bar{x}(k-1), \xi(k-1)\right) \coloneqq \phi_{\tilde{x}},$$

we are in the position to apply Theorem 1, whose assumptions are all satisfied. Indeed, the fact that the gradient and Hessian estimators are differentiable and Lipschitz continuous guarantees that assumption *(i)* holds true. Considering the boundary-layer system $\xi(k) = \varphi\left(\xi(k-1), \bar{x}\right)$, obtained by setting $\epsilon = 0$, we note that $\tilde{x}$ vanishes to zero and all the other components of ξ reach consensus. In particular, using (4) and (6) we see that the steady-state values of y, z are

$$y(t) = \hat{\nabla}^2 f(\bar{x}(t-1)) \odot \bar{x}(t-1) - \hat{\nabla} f(\bar{x}(t-1)),$$
$$z(t) = \hat{\nabla}^2 f(\bar{x}(t-1)).$$

This verifies assumption *(ii)* and ensures that $\Psi\left(\xi^*(\mathbb{0}), \mathbb{0}\right) = \mathbb{0}$. Noticing that $\phi\left(\bar{x}, \xi^\star(\bar{x})\right) = -\hat{\nabla} f(\bar{x}) \oslash \hat{\nabla}^2 f(\bar{x})$ and recalling Lemma 2, we have $\phi\left(\Gamma(\mu), \xi^\star(\Gamma(\mu))\right) = \mathbb{0}$. To satisfy assumption *(iii)* we assume without loss of generality that the target solution $\Gamma(\mu)$, which is arbitrarily close to the global minimum of $f(x)$, coincides with the origin. This corresponds to solving an equivalent problem in which the objective function is translated by an offset.

Finally, assumption *(iv)* can be satisfied by choosing as Lyapunov function $V(\bar{x}) = \left\|\hat{\nabla} f(\bar{x})\right\|^2$, which is always strictly positive except for $V(\mathbb{0}) = 0$. Indeed, it is possible to deduce the following bounds for $V(\bar{x})$, whose detailed derivations are contained in the technical report [15].

$$V(\bar{x}) \leq \left(L_1 + \frac{\mu\sqrt{d}L_2}{2}\right)^2 \|\bar{x} - \Gamma(\mu)\|^2,$$
$$V(\bar{x}) \geq \left(m^2 - 2L_1\frac{\mu^2 L_3}{6} - \left(\frac{\mu^2 L_3}{6}\right)^2\right) \|\bar{x} - \Gamma(\mu)\|^2,$$
$$\left\|\frac{\partial V}{\partial \bar{x}}\right\| \leq \left(2L_1 + \frac{\mu L_3 d}{3}\right)\left(L_1 + \frac{\mu\sqrt{d}L_2}{2}\right) \|\bar{x} - \Gamma(\mu)\|,$$
$$\frac{\partial V}{\partial \bar{x}}\phi\left(\bar{x}, \xi^\star(\bar{x})\right) \leq \left(\frac{2d\mu^2 L_3}{2m - L_3\mu^2} - \frac{12m}{12L_1 + L_3\mu^2}\right) \left(L_1 + \frac{\mu\sqrt{d}L_2}{2}\right)^2 \|\bar{x} - \Gamma(\mu)\|^2.$$

To ensure that the lower bound on $V(\bar{x})$ is not trivial and that we are moving along a descent direction, when $L_3 \neq 0$ we require that $\mu \leq \min\{\mu_1, \mu_2\}$, where $\mu_1 = \sqrt{\frac{6\left(\sqrt{L_1^2+m^2}-L_1\right)}{dL_3}}$, $\mu_2 = \sqrt{3\frac{\sqrt{4d^2L_1^2+m^2+4dL_1m+8m^2d}-2dL_1-m}{dL_3}}$. Since all the required conditions are verified, the proof is complete. ■

Remark 3: We recall that while the algorithm converges to $\Gamma(\mu)$, in virtue of Lemma 2 it is possible to make such point arbitrarily close to the actual global minimum of $f(x)$ by reducing μ. Therefore, the quantization parameter μ in the derivative estimators determines the accuracy of the solution reached by our algorithm.

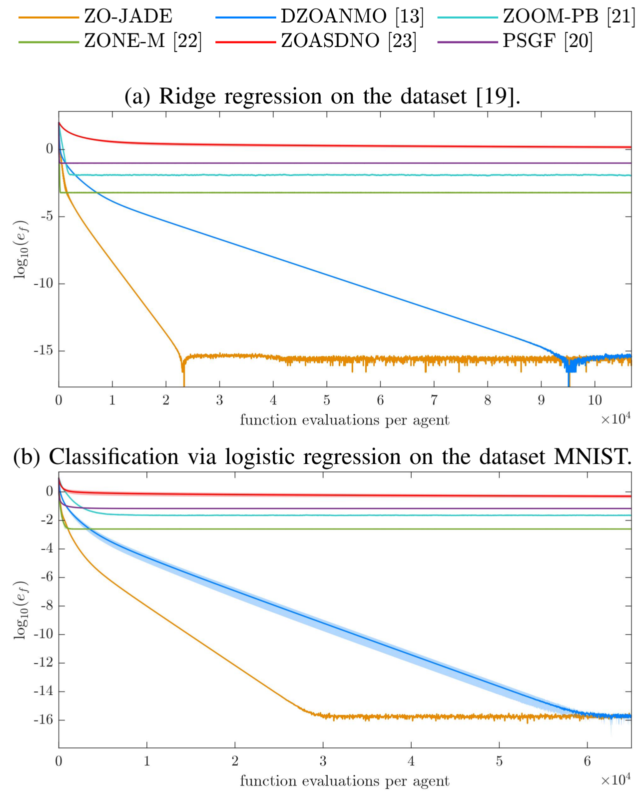

Fig. 1: Evolution of the loss function versus number of function queries. The plots show the average performance over multiple runs starting from different points, where the solid lines are the mean trajectories and the shaded areas represent the standard deviation.

Remark 4: Since the separation of time-scales is mainly an existential theorem, it is difficult to provide a bound on the parameter $\bar{\epsilon}$ or an explicit convergence rate. Even if this was possible, very conservative values of little practical use would be obtained. This is common to most works, where usually in the numerical simulations the hyperparameters are tuned by trial-and-error rather than according to possibly available theoretical bounds.

VI. Numerical results

Let us present some numerical results, which validate the effectiveness of ZO-JADE and show its superiority with respect to several state-of-the-art ZO algorithms for multi-agent optimization. In particular, we compare our method with the following distributed ZO algorithms: PSGF [20], which employs a push-sum strategy; the accelerated version of ZOOM, ZOOM-PB [21], which requires the Laplacian matrix of the network and uses a powerball function; ZONE-M [22], a primal-only version of the Method of Multipliers; the primal-dual method proposed in [23], dubbed here as ZOASDNO using the capital letters in the title of the paper, based on the Laplacian framework; the algorithm in [13] using a $2d$-points gradient estimator and gradient tracking, denoted here with the acronym DZOANMO. For all these algorithms, we tune the possible parameters as suggested in the respective papers. For ZOOM-PB and ZONE-M, whose randomized gradient estimators consider only a subset of coordinates, we test different numbers of search directions and show only the best performances obtained. We average over multiple runs, starting from different randomly generated initial points $x_i(0)$, which are the same for all the algorithms.

We consider a network of $n = 20$ agents satisfying Assumption 2, where the consensus matrix P is built using the Metropolis-Hastings weights, and we test the algorithms on two types of cost functions. In the first scenario we consider quadratic objectives, and specifically we perform ridge regression on the condition-based maintenance dataset [19] to predict the health state of a machine. In the second test we consider more general convex functions, addressing a binary classification problem via logistic regression. In particular we run one-vs-all classification, in which one target class needs to be distinguished from all the others, on the well-known MNIST dataset [24]. Adopting the approach followed by [25] on a similar dataset, we apply PCA to reduce the size of the problem to $d = 20$ and provide each agent with a balanced set $\mathcal{D}_i = \{s_1, \ldots, s_{|\mathcal{D}_i|}\}$, where half of the samples belong the target class and the other half to the remaining classes, equally mixed. The objective functions are the regularized log-loss cost functions

$$f_i(x, \mathcal{D}_i) = \frac{1}{\mathcal{D}_i} \sum_{k=1}^{|\mathcal{D}_i|} \log\left(1 + \exp\left(-l_k [s_k^T \ 1] x\right)\right) + \frac{w}{2} \|x\|^2 ,$$

where $l_k \in \{-1, 1\}$ is the label corresponding to the sample $s_k \in \mathbb{R}^{d-1}$ and the quadratic term with $w > 0$ ensures strong convexity. To take into account the possible differences between the local solutions x_i, we evaluate the performance of the algorithms using the loss function $e_f = \frac{\frac{1}{n}\sum_{i=1}^n f(x_i) - f(x^\star)}{|f(x^\star)|}$, where $x^\star$ is the unique minimizer of f. The speed of convergence is measured in terms of number of function evaluations, which in the zeroth-order optimization field are assumed to be the most expensive computations.

The numerical simulations clearly show that ZO-JADE outperforms the other algorithms in the considered scenarios, as it achieved an equivalent or lower loss value using a smaller amount of function evaluations. This suggests that our method is suitable both in the case where a very precise solution is desired and in the case where a limited amount of function queries are available.

VII. Conclusion

We presented a novel zeroth-order algorithm for distributed convex optimization, which is the first of its kind to exploit the second order derivative. We provided an intuitive explanation of our method and we formally proved its semi-global stability and exponentially fast convergence. The theoretical results are validated by numerical simulations on publicly available real-world datasets, in which ZO-JADE is shown to outperform several zeroth-order distributed algorithms present in the literature. We foresee interesting possibilities for future work, such as extensions to online and stochastic cases and the use of randomized derivative estimators.

References

[1] C. Audet and W. Hare, *Derivative-Free and Blackbox Optimization*, 01 2017.

[2] S. Liu, P.-Y. Chen, B. Kailkhura, G. Zhang, A. O. Hero III, and P. K. Varshney, "A primer on zeroth-order optimization in signal processing and machine learning: Principals, recent advances, and applications," *IEEE Signal Processing Magazine*, vol. 37, no. 5, pp. 43–54, 2020.

[3] J. Kiefer and J. Wolfowitz, "Stochastic estimation of the maximum of a regression function," *The Annals of Mathematical Statistics*, 1952.

[4] A. R. Conn, K. Scheinberg, and L. N. Vicente, *Introduction to Derivative-Free Optimization*. Society for Industrial and Applied Mathematics, 2009.

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

[6] J. Zhu, "Hessian inverse approximation as covariance for random perturbation in black-box problems," 2020. [Online]. Available: https://arxiv.org/abs/2011.13166

[7] B. Kim, H. Cai, D. McKenzie, and W. Yin, "Curvature-aware derivative-free optimization," *arXiv preprint arXiv:2109.13391*, 2021.

[8] A. S. Berahas, R. H. Byrd, and J. Nocedal, "Derivative-free optimization of noisy functions via quasi-Newton methods," *SIAM Journal on Optimization*, vol. 29, no. 2, pp. 965–993, 2019.

[9] K. Balasubramanian and S. Ghadimi, "Zeroth-order nonconvex stochastic optimization: Handling constraints, high-dimensionality and saddle-points," 2018. [Online]. Available: https://arxiv.org/abs/1809.06474

[10] A. R. Conn, K. Scheinberg, and L. N. Vicente, "Geometry of sample sets in derivative-free optimization: polynomial regression and underdetermined interpolation," *IMA journal of numerical analysis*, 2008.

[11] M. J. Powell, "The NEWUOA software for unconstrained optimization without derivatives," in *Large-scale nonlinear optimization*. Springer, 2006, pp. 255–297.

[12] L. Xiao, S. Boyd, and S.-J. Kim, "Distributed average consensus with least-mean-square deviation," *Journal of parallel and distributed computing*, vol. 67, no. 1, pp. 33–46, 2007.

[13] 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.

[14] A. Akhavan, M. Pontil, and A. Tsybakov, "Distributed zero-order optimization under adversarial noise," *Advances in Neural Information Processing Systems*, vol. 34, pp. 10 209–10 220, 2021.

[15] A. Maritan and L. Schenato, "ZO-JADE: Zeroth-order Curvature-Aware Multi-Agent Convex Optimization," *arXiv preprint arXiv:2303.07450*, 2023.

[16] D. Varagnolo, F. Zanella, A. Cenedese, G. Pillonetto, and L. Schenato, "Newton-Raphson consensus for distributed convex optimization," *IEEE Transactions on Automatic Control*, vol. 61, no. 4, 2015.

[17] D. Bertsekas and J. Tsitsiklis, *Parallel and distributed computation: numerical methods*. Athena Scientific, 2015.

[18] N. Bof, R. Carli, and L. Schenato, "Lyapunov theory for discrete time systems," *arXiv preprint arXiv:1809.05289*, 2018.

[19] A. Coraddu, L. Oneto, A. Ghio, S. Savio, D. Anguita, and M. Figari, "Machine learning approaches for improving condition based maintenance of naval propulsion plants," *Journal of Engineering for the Maritime Environment*, 2014.

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

[21] S. Zhang and C. Bailey, "Zeroth-order stochastic coordinate methods for decentralized non-convex optimization," *arXiv preprint arXiv:2204.04743*, 2022.

[22] D. Hajinezhad, M. Hong, and A. Garcia, "Zeroth order nonconvex multi-agent optimization over networks," *arXiv preprint arXiv:1710.09997*, 2017.

[23] X. Yi, S. Zhang, T. Yang, and K. H. Johansson, "Zeroth-order algorithms for stochastic distributed nonconvex optimization," *Automatica*, vol. 142, p. 110353, 2022.

[24] L. Deng, "The MNIST database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine*, 2012.

[25] N. D. Fabbro, S. Dey, M. Rossi, and L. Schenato, "A Newton-type algorithm for federated learning based on incremental Hessian eigenvector sharing," *arXiv preprint arXiv:2202.05800*, 2022.