Asynchronous Zeroth-Order Distributed Optimization with Residual Feedback

Yi Shen, Yan Zhang, Scott Nivison, Zachary I. Bell and Michael M. Zavlanos

***Abstract*— We consider a zeroth-order distributed optimization problem, where the global objective function is a black-box function and, as such, its gradient information is inaccessible to the local agents. Instead, the local agents can only use the values of the objective function to estimate the gradient and update their local decision variables. In this paper, we also assume that these updates are done asynchronously. To solve this problem, we propose an asynchronous zeroth-order distributed optimization method that relies on a one-point residual feedback to estimate the unknown gradient. We show that this estimator is unbiased under asynchronous updating, and theoretically analyze the convergence of the proposed method. We also present numerical experiments that demonstrate that our method outperforms two-point methods under asynchronous updating. To the best of our knowledge, this is the first asynchronous zeroth-order distributed optimization method that is also supported by theoretical guarantees.**

I. INTRODUCTION

Distributed optimization algorithms have been used to solve decision making problems in a wide range of application domains, including distributed machine learning [1], [2], resource allocation [3] and robotics [4], to name a few. In these problems, agents aim to find their local optimal decisions so that a global cost function that depends on the joint decisions is minimized. Existing algorithms, e.g., [5]–[8] often assume that the gradients of the objective function is known and available to the agents. However, this is not always the case in practice. For example, complex systems are often difficult to model explicitly [9], [10]. Similarly, in applications such as online marketing [11] and multi-agent games [12], the decisions of other agents cannot be observed and, therefore, the gradient of the objective function cannot be locally computed. Finally, many distributed optimization algorithms assume that all agents update their local decisions at the same time, which requires synchronization over the whole network and can be expensive to implement.

In this paper, we consider distributed optimization problems where a group of agents collaboratively minimize a common cost function that depends on their joint decisions. Moreover, we assume that the agents can only observe and update their local decision variables, and that the gradient of the common objective function with respect to each agent's local decision is not accessible. To solve this problem, zeroth-order optimization methods [13]–[16] have been proposed that estimate the gradient using the values of the objective function. Existing zeroth-order gradient estimators can be classified into two categories, namely, one-point feedback [13], [16] and two-point feedback [14], [15] estimators, depending on the number of decision points they query at each iteration. The first one-point gradient estimator is analyzed in [13] and has the form

$$G_\mu(x_k) = \frac{f(x_k + \mu u_k)}{\mu} u_k, \tag{1}$$

where μ is a smoothing parameter, u is sampled from a normal Gaussian distribution $\mathcal{N}(0, I_n)$ and I_n is an n dimensional identity matrix. The estimator (1) requires to evaluate the objective function at a single point $x_k + \mu u_k$ at iteration k but usually suffers large variance which slows down the optimization process. To reduce the variance of the one-point gradient estimator (1), the works in [14], [15] study the two-point gradient estimators

$$G_\mu(x_k) = \frac{f(x_k + \mu u_k) - f(x_k)}{\mu} u_k \tag{2}$$

$$\text{and} \quad G_\mu(x_k) = \frac{f(x_k + \mu u_k) - f(x_k - \mu u_k)}{2\mu} u_k, \tag{3}$$

that evaluate the objective function at two distinct decision points at iteration k. Recently, a new one-point gradient estimator has been proposed in [16], called the residual-feedback gradient estimator,

$$G_\mu(x_k) = \frac{f(x_k + \mu u_k) - f(x_{k-1} + \mu u_{k-1})}{\mu} u_k \tag{4}$$

that enjoys the same variance reduction effect of the two-point gradient estimators (2) and (3) but only requires to evaluate the objective function at a single decision point at each iteration. We note that verbatim application of the above centralized gradient estimators to the distributed problem considered in this paper requires synchronization across the agents. This is because the perturbation is according to the full decision vector and required to be implemented simultaneously. To the best of our knowledge, asynchronous zeroth-order distributed optimization methods have not been studied in the literature. If the objective function gradient is known, asynchronous distributed optimization methods with a common objective function have been studied in [17]–[19]. However, these works can not be directly extended to solve the black-box optimization problems considered here.

In this paper, we consider a model for asynchrony where a single agent is randomly activated at each time step to

*This work is supported in part by AFOSR under award #FA9550-19-1-0169 and by NSF under award CNS-1932011.

Yi Shen, Yan Zhang and Michael M. Zavlanos are with the Department of Mechanical Engineering and Materials Science, Duke University, Durham, NC, USA. Email: {yi.shen478, yan.zhang2, michael.zavlanos}@duke.edu

Scott Nivison and Zachary I. Bell are with the Air Force Research Laboratory, Eglin AFB, FL, USA. Email: {scott.nivison, zachary.bell.10}@us.af.mil.

query the objective function value at one decision point or update its local decision variable. Then, we propose an asynchronous zeroth-order distributed optimization algorithm that relies on extending the centralized residual-feedback gradient estimator (4) so that it can handle asynchronous queries and updates. Specifically, we show that the proposed zeroth-order gradient estimator provides an unbiased estimate of the gradient with respect to each agent's local decision. Also, we provide bounds on the second moment of this estimator, the first of their kind for any asynchronous zeroth-order gradient of this type, which we then use to show convergence of the proposed method. As expected, the variance of this estimator is greater than he variance of the estimator in (4), since other agents in the network can update their decision variables between an agent's two consecutive queries of the objective function.

We note that, almost concurrently with this work [20] proposed an asynchronous zeroth-order optimization algorithm that relies on the two-point gradient estimator (3). Unlike the method proposed here, [20] assumes that when a single agent queries the values of the objective function at decision points $x_k + \mu u_k$ and x_k (or $x_k - \mu u_k$), the other agents in the network cannot update their local decision variables even if they are activated. This assumption limits the number of updates the agents can make during a fixed period of time and affects the performance of the asynchronous system. Related is also work on distributed zeroth-order methods for the optimization of functions that are the sum of local objective functions, see [21]–[24]. However, these methods assume synchronous updates.

The rest of this paper is organized as follows. In Section II, we formulate the problem under consideration and present preliminary results on zeroth-order gradient estimators. In Section III, we present the proposed asynchronous zeroth-order distributed optimization algorithm with residual feedback gradient estimation, and analyze its convergence. In Section IV, we numerically validate the proposed algorithm and in Section V we conclude the paper.

II. PROBLEM FORMULATION AND PRELIMINARIES

Consider a multi-agent system consisting of N agents that collaboratively solve the unconstrained optimization problem

$$\min_{x} \; f(x), \tag{5}$$

where the cost function f is non-convex and smooth, $x := (x^1, \dots, x^N) \in \mathbb{R}^n$ is the joint decision vector, and $x^i \in \mathbb{R}^{n_i}$ is the local decision vector of agent $i \in \{1, \dots, N\}$. We first make the following assumptions on the cost function f.

Assumption 1. *The cost function $f(x) : \mathbb{R}^n \to \mathbb{R}$ is bounded below by f^*. It is L_0-Lipschtiz and L_1-smooth, i.e.,*

$$|f(x) - f(y)| \leq L_0 \|x - y\|,$$
$$\|\nabla f(x) - \nabla f(y)\| \leq L_1 \|x - y\|,$$

for all $x, y \in \mathbb{R}^n$.

As shown in [14], L_1-smoothness is equivalent to the condition;

$$|f(y) - f(x) - \langle \nabla f(x), y - x \rangle| \leq \frac{1}{2} L_1 \|x - y\|^2, \tag{6}$$

for all $x, y \in \mathbb{R}^n$. For each agent i, define the local smoothing function:

$$f_{\mu_i}(x) = \frac{1}{\kappa_i} \int f(x + \mu_i u_i) e^{-\frac{1}{2}\|u_i\|^2} du_i^i, \tag{7}$$

where $\kappa_i = \int e^{-\frac{1}{2}\|u_i\|^2} du_i^i$. The random sampling vector $u_i = \{u_i^1, \dots, u_i^N\} \in \mathbb{R}^n$ is a vector of all zeros except for the entry u_i^i that is sampled from $\mathcal{N}(0, I_{n_i})$.[1] Note that f_{μ_i} preserves all the Lipschitz conditions of f as proved in [14]. Specifically, we have the following lemma.

Lemma 1. *Under Assumption 1, we have that , for all agents i, $f_{\mu_i}(x) : \mathbb{R}^n \to \mathbb{R}$ is L_0-Lipschtiz and L_1-smooth.*

As a result, (6) also holds for f_{μ_i}, which allows us to bound the approximation errors of f_{μ_i} and $\nabla_i f_{\mu_i}$ with respect to (w.r.t.) to f and $\nabla_i f$, as shown in Lemma 2 below that is adopted from [14]. In Lemma 2 and the following analysis, we denote by $\nabla f(x) \in \mathbb{R}^n$ the gradient of $f(x)$. Moreover, we define by $\nabla_i f(x) \in \mathbb{R}^n$ the projection of $\nabla f(x)$ onto the index i by setting the entries of $\nabla f(x)$ not equal to i to be 0.

Lemma 2. *Under Assumption 1, the cost function f and its corresponding smoothed function f_{μ_i} satisfy $\forall i \in \{1, \dots, N\}$*

$$|f_{\mu_i}(x) - f(x)| \leq \frac{\mu_i^2}{2} L_1 n_i, \tag{8}$$

$$\|\nabla_i f_{\mu_i}(x) - \nabla_i f(x)\| \leq \frac{\mu_i}{2} L_1 (n_i + 3)^{3/2}. \tag{9}$$

Next, we define the model that the agents use to asynchronously update their decision variables.

Definition 1 (Asynchrony Model)**.** *At each time step, one agent is independently and randomly selected according to a fixed distribution $P = [p_1, \dots, p_N]$. The selected agent i can query the value of the cost function[2] once and update its local decision variable, while the decisions of the other agents $\{x^j\}_{j \neq i}$ are fixed. The agents only communicate with a central entity that has access to the global cost function but not with each other.*

Remark 1. *Definition 1 can be satisfied when all agents make queries and update their decisions according to their local clock without any central coordination. Specifically, let the time interval between each agent's consecutive queries be called a waiting time. Then, if each local waiting time is random and subject to an exponential distribution, according to Chapter 2.1 in [25], Definition 1 will be satisfied.*

[1] In the following analysis, we drop the agent index i in u_i for simplicity.

[2] Here, we assume that each agent receives noiseless feedback $f(x)$. The proposed method can be extended to noisy feedback with bounded variance.

III. ALGORITHM DESIGN AND ANALYSIS

In this section, we present the proposed asynchronous zeroth-order distributed optimization algorithm and analyze its convergence rate. To do so, we first propose an asynchronous zeroth-order gradient estimator based on the centralized residual feedback estimator (4). For every agent i, at time step k, this estimator takes the form

$$G_{\mu_i}(x_k) = \frac{f(x_k + \mu_i u_k) - f(x_{k-M} + \mu_i u_{k-M})}{\mu_i} u_k, \quad (10)$$

where $k - M$ is a random index denoting the iteration when agent i conducted its most recent update. This index takes values on a global time scale. The random sampling vector u_k is as defined in (7). Note that G_{μ_i} is different from the centralized zeroth-order gradient estimator (4), where u_k is a perturbation along the full decision vector x. Here, G_{μ_i} estimates the gradient by perturbing the function f along a random direction restricted to agent i's block of the full decision vector x and uses the previous query to reduce the variance. Indeed, G_{μ_i} provides an unbiased gradient estimate of the corresponding smoothed function f_{μ_i} restricted to agent i's block, as shown in the following lemma.

Lemma 3. *For each agent i, we have that*

$$\mathbb{E}\left[G_{\mu_i}(x_k)\right] = \nabla_i f_{\mu_i}(x_k).$$

Proof. Taking the expectation of both sides of (10), we obtain that

$$\begin{aligned}\mathbb{E}\left[G_{\mu_i}(x_k)\right] &= \mathbb{E}\left[\frac{f(x_k + \mu_i u_k) - f(x_{k-M} + \mu_i u_{k-M})}{\mu_i} u_k\right] \\ &= \mathbb{E}\left[\frac{f(x + \mu_i u_k)}{\mu_i} u_k\right] = \nabla_i f_{\mu_i}(x_k),\end{aligned}$$

where the second equality follows from the fact that x_{k-M} and u_{k-M} are independent from u_k. The last equality follows from the definitions of f_{μ_i} and $\nabla_i f_{\mu_i}$. □ □

Remark 2. *Note that both $G_{\mu_i}(x_k)$ and $\nabla_i f_{\mu_i}$ are vectors in $\mathbb{R}^n$ with entries equal to zero at blocks other than i.*

Using the local gradient estimate $G_{\mu_i}(x_k)$, we can define the update rule for every agent i as

$$x_{k+1} = x_k - \alpha_i G_{\mu_i}(x_k), \quad (11)$$

where α_i is the step size. The proposed asynchronous zeroth-order distributed optimization algorithm with residual feedback is described in Algorithm 1[3].

Without loss of generality, we assume that decision variables $x^i \in \mathbb{R}^{\bar{n}}$ of all agents i have the same dimensions, and that the step sizes and smoothing parameters of all agents are also the same, i.e., $\alpha_i = \alpha$ and $\mu_i = \mu$. To analyze the convergence of Algorithm 1, we need to bound the second moment of the proposed gradient estimator $G_{\mu_i}(x_k)$.

[3]Note that, we can also extend (2) for asynchronous problems and design the algorithm thereof. The lemmas and theorems proved in this paper can be easily adapted to this case as well. In Section IV, we will compare these two gradient estimators empirically. However, the extension of (3) is non-trivial. It can be verified that Lemma 3 does not hold for the extension of (3). We leave it as future work.

Algorithm 1 Asynchronous Zeroth-Order Residual Feedback

Require: sampling rate p_i with $\sum_{i=1}^N p_i = 1$, decision variable x_0^i, smoothing parameter μ_i and step size α_i for all agents i. Set the iteration counter $t = 0$ and let T be the maximum number of iterations.

1: **for** $t \leq T$ **do**
2: sample an index i_t according to $\mathbb{P}(i_t = i) = p_i$
3: sample $u^i \sim \mathcal{N}(0, I_{n_i})$
4: query the function value $f(x + \mu_i u)$
5: compute G_{μ_i} according to (10)
6: update local decision $x^i \leftarrow x^i - \alpha_i G_{\mu_i}(x^i)$
7: update the time step counter $t \leftarrow t + 1$
8: **end for**

However, under the asynchronous framework considered in this paper, there can be a random number of agents updating their local decision variables between the two queries made by agent i at time steps k and $k - M$ in (10). These updates by other agents introduce additional variance into the estimator (10) compared to the variance of the centralized estimator analyzed in [16]. Next, we analyze the effect of the asynchronous updates on the second moment of the zeroth-order gradient estimator. To the best of our knowledge, this is the first time that a bound on the second moment of a zeroth-order gradient estimator is provided for asynchronous problems. An additional contribution of this work, is that the proof technique presented below can be extended to obtain similar results for the two-point gradient estimator (2).

Lemma 4. *Let Assumptions 1 hold under the Asynchrony Model, and define by $\mathbb{E}[\|G_{\tilde{\mu}}(x_k)\|^2] := \mathbb{E}_{i_k}[\mathbb{E}_{u_{[k]}, i_{[k-1]}}[\|G_{\mu_i}(x_k)\|^2 | i_k = i]]$, where $u_{[k]} = (u_1, \ldots, u_k)$ and $i_{[k]} = (i_1, \ldots, i_k)$, Then, running the asynchronous Algorithm 1, we have that $\mathbb{E}[\|G_{\tilde{\mu}}(x_k)\|^2]$ satisfies*

$$\begin{aligned}&\mathbb{E}\left[\|G_{\tilde{\mu}}(x_k)\|^2\right] \\ &\leq \frac{2\bar{n}L_0^2\alpha^2 k}{\mu^2} \sum_{m=0}^{k-1} (1 - p_{\min})^m \mathbb{E}\left[\|G_{\tilde{\mu}}(x_{k-m-1})\|^2\right] \\ &\quad + 4L_0^2\left((4 + \bar{n})^2 + \bar{n}^2\right),\end{aligned} \quad (12)$$

where $p_{\min} = \min_i p_i$, and the expectations are taken w.r.t. the sequence of random exploration directions $\{u_k\}$ and the sequence of random indices of activated agents $\{i_k\}$.

Proof. Suppose that at time step k, agent i is selected. Taking the second moment of G_{μ_i} and using equation (10), we we have[4]

$$\begin{aligned}&\mathbb{E}_{u_{[k]}, i_{[k-1]}}\left[\|G_{\mu_i}(x_k)\|^2 | i_k = i\right] \leq \\ &\mathbb{E}\left[\frac{\left(f(x_k + \mu_i u_k) - f(x_{k-M} + \mu_i u_{k-M})\right)^2 \|u_k\|^2}{\mu_i^2}\right]\end{aligned} \quad (13)$$

[4]To simplify the notation, when it is clear from the context, we drop the subscript of the expectation and the conditional event, e.g., $\mathbb{E}[\|G_{\mu_i}(x_k)\|^2] := \mathbb{E}_{u_{[k]}, i_{[k-1]}}[\|G_{\mu_i}(x_k)|i_k = i\|^2]$.

Notice that

$$\begin{aligned}&\left(f(x_k+\mu_i u_k)-f(x_{k-M}+\mu_i u_{k-M})\right)^2\\&\leq 2\underbrace{\left(f(x_k+\mu_i u_k)-f(x_{k-M}+\mu_i u_k)\right)^2}_{\text{a}}\\&+2\underbrace{\left(f(x_{k-M}+\mu_i u_k)-f(x_{k-M}+\mu_i u_{k-M})\right)^2}_{\text{b}}.\end{aligned}\tag{14}$$

Substituting (14) into (13), we obtain that

$$\begin{aligned}&\mathbb{E}_{u_{[k]},i_{[k-1]}}\left[\left\|G_{\mu_i}(x_k)\right\|^2|i_k=i\right]\\&\leq\mathbb{E}\left[\frac{2a+2b}{\mu_i^2}\left\|u_k\right\|^2\right]\\&\leq\mathbb{E}\left[\frac{2L_0^2\left\|x_k-x_{k-M}\right\|^2}{\mu_i^2}\left\|u_k\right\|^2\right]+\mathbb{E}\left[\frac{2b}{\mu_i^2}\left\|u_k\right\|^2\right]\\&\leq\frac{2L_0^2\bar{n}}{\mu_i^2}\mathbb{E}\left[\left\|x_k-x_{k-M}\right\|^2\right]+\mathbb{E}\left[\frac{2b}{\mu_i^2}\left\|u_k\right\|^2\right],\end{aligned}\tag{15}$$

where the second inequality holds due to Lipschitzness of f and the last inequality holds since x_k-x_{k-M} is independent from u_k and $\mathbb{E}[\|u_k\|^2]=\bar{n}$. We first bound the second term in the right-hand-side of (15). Specifically, we have that

$$\begin{aligned}&\mathbb{E}\left[\frac{2b}{\mu_i^2}\left\|u_k\right\|^2\right]\leq\mathbb{E}\left[2L_0^2\left\|u_k-u_{k-M}\right\|^2\left\|u_k\right\|^2\right]\\&\leq\mathbb{E}\left[4L_0^2\left(\left\|u_k\right\|^2+\left\|u_{k-M}\right\|^2\right)\left\|u_k\right\|^2\right]\\&\leq\mathbb{E}\left[4L_0^2\left\|u_k\right\|^4\right]+\mathbb{E}\left[4L_0^2\left\|u_{k-M}\right\|^2\right]\mathbb{E}\left[\left\|u_k\right\|^2\right]\\&\leq 4L_0^2\left((4+\bar{n})^2+\bar{n}^2\right),\end{aligned}\tag{16}$$

where the first inequality holds due to Lipschitzness of f, the third inequality holds since u_{k-M} is independent from u_k and the last inequality follows from Lemma 1 in [14].

Next, we bound the first term in the right-hand-side of (15) containing the second moment of x_k-x_{k-M}. Given that agent i updates at time step k, we can partition the sequence of all past updates into k events $A_m^i=\{M=m\}$, where A_m^i represents all sequences of updates such that the most recent update by agent i is at global time step $k-m$. In particular, A_k^i indicates that agent i has not been updated before and k is the first time that this agent gets updated. It is easy to see that the sets $\{A_m^i\}_{m=1}^k$ are disjoint and contain all possible sequences of updates by the team of agents. Using the definition of these events, we can rewrite the conditional expectation of $\|x_k-x_{k-M}\|^2$ as

$$\begin{aligned}Y_i&:=\mathbb{E}_{u_{[k]},i_{[k-1]}}\left[\left\|x_k-x_{k-M}\right\|^2|i_k=i\right]\\&=\sum_{m=1}^k\mathbb{E}_{u_{[k]}}\left[\left\|x_k-x_{k-m}\right\|^2|A_m^i,i_k=i\right]\mathbb{P}(A_m^i).\end{aligned}\tag{17}$$

Equation (17) can be rewritten as

$$\begin{aligned}Y_i&=\sum_{m=1}^k\mathbb{E}\left[\left\|x_k-x_{k-m}\right\|^2|A_m^i\right]\mathbb{P}(A_m^i)\\&=\sum_{m=1}^k\mathbb{E}\left[\left\|\sum_{l=0}^{m-1}(x_{k-l}-x_{k-l-1})\right\|^2|A_m^i\right]\mathbb{P}(A_m^i)\\&\leq\sum_{m=1}^k\mathbb{E}\left[m\sum_{l=0}^{m-1}\left\|(x_{k-l}-x_{k-l-1})\right\|^2|A_m^i\right]\mathbb{P}(A_m^i),\end{aligned}$$

where the last inequality holds due to the fact that $\left(\sum_{m=1}^k a_m\right)^2\leq k\sum_{m=1}^k a_m^2$. We first collect all the terms containing $\|x_k-x_{k-1}\|^2$, which we denote by $\{Y_i:\|x_k-x_{k-1}\|^2\}$. Then, we have that

$$\begin{aligned}&\{Y_i:\|x_k-x_{k-1}\|^2\}=\sum_{m=1}^k m\mathbb{E}\left[\left\|x_k-x_{k-1}\right\|^2|A_m^i\right]\mathbb{P}(A_m^i)\\&\leq k\sum_{m=1}^k\mathbb{E}\left[\left\|x_k-x_{k-1}\right\|^2|A_m^i\right]\mathbb{P}(A_m^i)\\&=k\mathbb{E}\left[\left\|x_k-x_{k-1}\right\|^2\right],\end{aligned}\tag{18}$$

where the last equation holds by the definition of conditional expectation. Next we collect all the terms containing $\|x_{k-s}-x_{k-s-1}\|^2$ for all $s\in\{1,\dots,k-1\}$. Specifically, we have that

$$\begin{aligned}&\{Y_i:\|x_{k-s}-x_{k-s-1}\|^2\}\\&=\sum_{m=s+1}^k m\mathbb{E}\left[\left\|x_{k-s}-x_{k-s-1}\right\|^2|A_m^i\right]\mathbb{P}(A_m^i)\\&\leq k\sum_{m=s+1}^k\mathbb{E}\left[\left\|x_{k-s}-x_{k-s-1}\right\|^2|A_m^i\right]\mathbb{P}(A_m^i).\end{aligned}\tag{19}$$

We claim that the right hand side of (19) satisfies the following equation

$$\begin{aligned}&k\sum_{m=s+1}^k\mathbb{E}\left[\left\|x_{k-s}-x_{k-s-1}\right\|^2|A_m^i\right]\mathbb{P}(A_m^i)\\&=k\mathbb{P}(A_{1:s}^{ic})\mathbb{E}\left[\left\|x_{k-s}-x_{k-s-1}\right\|^2\right],\end{aligned}\tag{20}$$

where $A_{1:s}^{ic}=\cup_{m=s+1}^k A_m^i$. To see this, we first observe that

$$\begin{aligned}&\mathbb{E}\left[\left\|x_{k-s}-x_{k-s-1}\right\|^2\right]\\&=\sum_{m=1}^k\mathbb{E}\left[\left\|x_{k-s}-x_{k-s-1}\right\|^2|A_m^i\right]\mathbb{P}(A_m^i)\\&=\mathbb{E}\left[\left\|x_{k-s}-x_{k-s-1}\right\|^2|A_{1:s}^i\right]\mathbb{P}(A_{1:s}^i)\\&\quad+\mathbb{E}\left[\left\|x_{k-s}-x_{k-s-1}\right\|^2|A_{1:s}^{ic}\right]\mathbb{P}(A_{1:s}^{ic}),\end{aligned}\tag{21}$$

where $A_{1:s}^i=\cup_{m=1}^s A_m^i$ and $A_{1:s}^{ic}$ is the complement of $A_{1:s}^i$. The second equality follows from the property of conditional expectation of disjoints events. Note that

$\mathbb{E}[\|x_{k-s} - x_{k-s-1}\|^2 | A^i_{1:s}]$ and $\mathbb{E}[\|x_{k-s} - x_{k-s-1}\|^2 | A^{ic}_{1:s}]$ are equal. Specifically, event $A^i_{1:s}$ and event $A^{ic}_{1:s}$ only differ after time step $k-s$, where $A^i_{1:s}$ contains all sequences of updates where i update after $k-s$ and $A^{ic}_{1:s}$ contains all sequences of updates where i does not updates after $k-s$. Since both events $A^i_{1:s}$ and $A^{ic}_{1:s}$ do not affect the agents' updates before time step $k-s$, we have that

$$\mathbb{E}\left[\|x_{k-s} - x_{k-s-1}\|^2\right] = \mathbb{E}\left[\|x_{k-s} - x_{k-s-1}\|^2 | A^{ic}_{1:s}\right].$$

Combining the above equality with (19), we have that

$$\begin{aligned} &k \sum_{m=s+1}^{k} \mathbb{E}\left[\|x_{k-s} - x_{k-s-1}\|^2 | A^i_m\right] \mathbb{P}(A^i_m) \\ &= k\mathbb{P}(A^{ic}_{1:s})\mathbb{E}\left[\|x_{k-s} - x_{k-s-1}\|^2 | A^{ic}_{1:s}\right] \\ &= k\mathbb{P}(A^{ic}_{1:s})\mathbb{E}\left[\|x_{k-s} - x_{k-s-1}\|^2\right], \end{aligned}$$

which completes the proof of (20). Then, we can bound Y_i in (17) by combining (18) and (20) and have that

$$\begin{aligned} \mathbb{E}\left[\|x_k - x_{k-M}\|^2\right] &\le k\mathbb{E}\left[\|x_k - x_{k-1}\|^2\right] \\ &+ k\sum_{m=1}^{k-1} \mathbb{P}(A^{ic}_{1:m})\mathbb{E}\left[\|x_{k-m} - x_{k-m-1}\|^2\right]. \end{aligned} \tag{22}$$

By definition, we have $\mathbb{P}(A^i_m) = p_i(1-p_i)^{m-1}$ for $1 \le m < k$, and $\mathbb{P}(A^i_k) = (1-p_i)^k$ for $m = k$, where p_i is the probability of agent i being sampled at each time step. As a result, $\mathbb{P}(A^{ic}_{1:m}) = (1-p_i)^m$. Substituting these probabilities into (22), we have that

$$\begin{aligned} &\mathbb{E}\left[\|x_k - x_{k-M}\|^2\right] \\ &\le k\sum_{m=0}^{k-1}(1-p_{\min})^m \mathbb{E}\left[\|x_{k-m} - x_{k-m-1}\|^2\right], \end{aligned} \tag{23}$$

where $p_{\min} = \min_i p_i$. Substituting (23) and (16) into (15), we get that

$$\begin{aligned} &\mathbb{E}\left[\|G_{\mu_i}(x_k)\|^2\right] \\ &\le \frac{2L_0^2\bar{n}k}{\mu_i^2}\sum_{m=0}^{k-1}(1-p_{\min})^m \mathbb{E}\left[\|x_{k-m} - x_{k-m-1}\|^2\right] \\ &\quad + 4L_0^2\left((4+\bar{n})^2 + \bar{n}^2\right). \end{aligned} \tag{24}$$

Recall that all expectations from the beginning of the proof are taken conditioned on the event $\{i_k = i\}$. Now, taking the expectation w.r.t. i_k on both sides of (24), and substituting the step size α and smoothing parameter μ into (24), we get that

$$\begin{aligned} &\mathbb{E}\left[\|G_{\tilde{\mu}}(x_k)\|^2\right] \\ &\le \frac{2\bar{n}L_0^2\alpha^2 k}{\mu^2}\sum_{m=0}^{k-1}(1-p_{\min})^m \mathbb{E}\left[\|G_{\tilde{\mu}}(x_{k-m-1})\|^2\right] \\ &\quad + 4L_0^2\left((4+\bar{n})^2 + \bar{n}^2\right), \end{aligned}$$

where $\mathbb{E}[\|x_{k-m} - x_{k-m-1}\|^2] = \alpha^2\mathbb{E}[\|G_{\tilde{\mu}}(x_{k-m-1})\|^2]$ according to the update rule (11) and the definition of $\mathbb{E}[\|G_{\tilde{\mu}}\|^2]$ as in Lemma 4. The proof is complete. □ □

Lemma 4 indicates that the second moment of the zeroth-order gradient estimate at time step k is related to the second moments of all previous gradient estimates. Specifically, the effect of the second moment of the past gradient estimates on the current estimate diminishes geometrically over time. Next, using Lemma 4, we present a bound on the accumulated second moments of the residual-feedback gradient estimates from $k = 0$ to $T-1$, which we will later use to prove our main theorem.

Lemma 5. *Let Assumptions 1 hold under the Asynchrony Model. Then, running the asynchronous updates Algorithm 1, we have that*

$$\begin{aligned} \sum_{k=0}^{T-1}\mathbb{E}\left[\|G_{\tilde{\mu}}(x_k)\|^2\right] &\le \frac{1-\beta}{1-(\gamma+\beta)}\mathbb{E}\left[\|G_{\tilde{\mu}}(x_0)\|^2\right] \\ &+ (T-1)\frac{1-\beta}{1-(\gamma+\beta)}M - \frac{\gamma}{(1-(\gamma+\beta))^2}M, \end{aligned}$$

where $\gamma = \frac{2\bar{n}L_0^2\alpha^2(T-1)}{\mu^2}$, $\beta = 1 - p_{\min}$, $M = 4L_0^2\left((4+\bar{n})^2 + \bar{n}^2\right)$ *and provided with* $0 < \gamma + \beta < 1$.

The proof follows from Lemma 6 in the Appendix.

Theorem 1. *Let Assumptions 1 hold under the Asynchrony Model. Moreover, run the asynchronous algorithm Algorithm 1 for T iterations and let $\tilde{x}$ be uniformly randomly selected from T iterations. Then, selecting the step size $\alpha = \frac{\sqrt{p_{\min}}}{T^{\frac{2}{3}}}$ and the smoothing parameter $\mu = \frac{2L_0\sqrt{\bar{n}}}{T^{\frac{1}{6}}}$, we have $\mathbb{E}\left[\|\nabla f(\tilde{x})\|^2\right] \le \mathcal{O}(\bar{n}^3 T^{-\frac{1}{3}})$.*

Proof. Substituting x_{k+1} and x_k in the version of (6) for the smoothed function f_{μ_i}, we obtain that

$$\begin{aligned} &f_{\mu_i}(x_{k+1}) \\ &\le f_{\mu_i}(x_k) + \langle \nabla f_{\mu_i}(x_k), x_{k+1} - x_k\rangle + \frac{L_1}{2}\|x_{k+1} - x_k\|^2 \\ &= f_{\mu_i}(x_k) - \alpha\langle \nabla_i f_{\mu_i}(x_k), \Delta_{i,k}\rangle - \alpha\|\nabla_i f_{\mu_i}(x_k)\|^2 \\ &\quad + \frac{L_1\alpha^2}{2}\|G_{\mu_i}(x_k)\|^2, \end{aligned} \tag{25}$$

where $\Delta_{i,k} := G_{\mu_i}(x_k) - \nabla_i f_{\mu_i}(x_k)$. The first equality follows by (11) and the fact that $\langle \nabla f_{\mu_i}(x_k), x_{k+1} - x_k\rangle = \langle \nabla_i f_{\mu_i}(x_k), x_{k+1} - x_k\rangle$, which holds since x_{k+1} and x_k only differ at block i. Taking expectation w.r.t. $u_{[k]}$ and $i_{[k-1]}$ on both sides of (25) conditioned on the event $\{i_k = i\}$, we get that

$$\begin{aligned} \mathbb{E}\left[\|\nabla_i f_{\mu_i}(x_k)\|^2\right] &\le \frac{\mathbb{E}\left[f_{\mu_i}(x_k)\right] - \mathbb{E}\left[f_{\mu_i}(x_{k+1})\right]}{\alpha} \\ &+ \frac{L_1\alpha}{2}\mathbb{E}\left[\|G_{\mu_i}(x_k)\|^2\right], \end{aligned} \tag{26}$$

where the inner-product term $\langle \nabla_i f_{\mu_i}(x_k), \Delta_{i,k}\rangle$ disappears since $\mathbb{E}[\Delta_{i,k}] = 0$. According to Lemma 2 and using the fact that $(a+b)^2 \le 2a^2 + 2b^2$, we have

$$\|\nabla_i f(x)\|^2 \le 2\|\nabla_i f_{\mu_i}(x)\|^2 + \mu_i^2 {L_1}^2(n_i+3)^3. \tag{27}$$

Combining (26) and (27) and, we obtain that

$$\frac{1}{2}\mathbb{E}\left[\|\nabla_i f(x_k)\|^2\right] \leq \frac{\mathbb{E}\left[f_{\mu_i}(x_k)\right] - \mathbb{E}\left[f_{\mu_i}(x_{k+1})\right]}{\alpha} + \frac{L_1\alpha}{2}\mathbb{E}\left[\|G_{\mu_i}(x_k)\|^2\right] + \frac{1}{2}\mu^2 {L_1}^2(\bar{n}+3)^3, \tag{28}$$

where the last term follows by substituting the common smoothing parameter μ and agents' dimension $\bar{n}$. Taking expectation on both sides of (28) w.r.t. i_k, we have that

$$\frac{1}{2}\mathbb{E}_{i_k}\left[\mathbb{E}_{u_{[k]},i_{[k-1]}}\left[\|\nabla_i f(x_k)\|^2 \,|i_k = i\right]\right] \leq \frac{\mathbb{E}\left[f_{\tilde{\mu}}(x_k)\right] - \mathbb{E}\left[f_{\tilde{\mu}}(x_{k+1})\right]}{\alpha} + \frac{L_1\alpha}{2}\mathbb{E}\left[\|G_{\tilde{\mu}}(x_k)\|^2\right] + \frac{1}{2}\mu^2 {L_1}^2(\bar{n}+3)^3, \tag{29}$$

where $\mathbb{E}[f_{\tilde{\mu}}(x_k)] := \mathbb{E}_{i_k}[\mathbb{E}_{u_{[k]},i_{[k-1]}}[f_{\mu_{i_k}}(x_k)|i_k = i]]$ and $\mathbb{E}[f_{\tilde{\mu}}(x_{k+1})] := \mathbb{E}_{i_k}[\mathbb{E}_{u_{[k]},i_{[k-1]}}[f_{\mu_{i_k}}(x_{k+1})|i_k = i]]$. $\mathbb{E}[\|G_{\tilde{\mu}}(x_k)\|^2]$ follows from the definition in Lemma 4. Next, we show that the left hand side of (29) satisfies

$$\mathbb{E}_{i_k}[\mathbb{E}_{u_{[k]},i_{[k-1]}}[\|\nabla_{i_k} f(x_k)\|^2 \,|i_k = i]] \geq p_{\min}\mathbb{E}_{u_{[k]},i_{[k]}}\|\nabla f(x_k)\|^2. \tag{30}$$

To see this, by definitions of the projected gradient as in Section II, since $\nabla_i f(x_k)$ is only nonzero at block i, we have $\|\nabla f(x_k)\|^2 = \sum_{i=1}^N \|\nabla_i f(x_k)\|^2$. Therefore, we can further get that

$$\mathbb{E}_{u_{[k]},i_{[k]}}\|\nabla f(x_k)\|^2 = \sum_{i=1}^N \mathbb{E}_{u_{[k]},i_{[k]}}\|\nabla_i f(x_k)\|^2 = \sum_{i=1}^N \mathbb{E}_{u_{[k]},i_{[k-1]}}[\|\nabla_i f(x_k)\|^2 \,|i_k = i], \tag{31}$$

where the second equality holds since x_k is independent from i_k. Therefore, according to (31), to show the inequality (30), it is sufficient to show $\mathbb{E}_{i_k}[\mathbb{E}_{u_{[k]},i_{[k-1]}}[\|\nabla_{i_k} f(x_k)\|^2 \,|i_k = i]] \geq p_{\min}\sum_{i=1}^N \mathbb{E}_{u_{[k]},i_{[k-1]}}[\|\nabla_i f(x_k)\|^2 \,|i_k = i]$. This is simple to prove because $\mathbb{E}_{i_k}[\mathbb{E}_{u_{[k]},i_{[k-1]}}[\|\nabla_{i_k} f(x_k)\|^2 \,|i_k = i]] = \sum_i p_i \mathbb{E}_{u_{[k]},i_{[k-1]}}[\|\nabla_i f(x_k)\|^2 \,|i_k = i]$. Therefore, inequality (30) is true. Substituting (30) into (29) and then summing (29) from $k = 0$ to $T-1$, we have that

$$\frac{p_{\min}}{2}\sum_{k=0}^{T-1}\mathbb{E}\left[\|\nabla f(x_k)\|^2\right] \leq \frac{\mathbb{E}\left[f_{\tilde{\mu}}(x_0)\right] - \mathbb{E}\left[f_{\tilde{\mu}}(x_T)\right]}{\alpha} + \sum_{k=0}^{T-1}\frac{L_1\alpha}{2}\mathbb{E}\left[\|G_{\tilde{\mu}}(x_k)\|^2\right] + \frac{\mu^2}{2}{L_1}^2(\bar{n}+3)^3 T. \tag{32}$$

Applying Lemma 5 to (32), we get that

$$\frac{p_{\min}}{2}\sum_{k=0}^{T-1}\mathbb{E}\left[\|\nabla f(x_k)\|^2\right] \leq \frac{\mathbb{E}\left[f_{\tilde{\mu}}(x_0)\right] - \mathbb{E}\left[f_{\tilde{\mu}}(x_T)\right]}{\alpha} + \frac{L_1\alpha}{2}(T-1)\frac{1-\beta}{1-(\gamma+\beta)}M + \frac{L_1\alpha}{2}\frac{1-\beta}{1-(\gamma+\beta)}\mathbb{E}\left[\|G_{\tilde{\mu}}(x_0)\|^2\right] + \frac{\mu^2}{2}{L_1}^2(\bar{n}+3)^3 T - \frac{L_1\alpha}{2}\frac{\gamma}{(1-(\gamma+\beta))^2}M, \tag{33}$$

where γ, β and M are as defined in Lemma 5. Selecting $\mu = \frac{2L_0}{T^{\frac{1}{6}}}$ and $\alpha = \frac{\sqrt{p_{\min}}}{\sqrt{\bar{n}}T^{\frac{2}{3}}}$, we have $\gamma \leq \frac{p_{\min}}{2}$ and $1-(\gamma+\beta) \geq \frac{p_{\min}}{2}$. Substituting these values into (33) and omitting the negative term, we obtain that

$$\frac{p_{\min}}{2}\sum_{k=0}^{T-1}\mathbb{E}\left[\|\nabla f(x_k)\|^2\right] \leq \frac{\mathbb{E}\left[f_{\tilde{\mu}}(x_0)\right] - f^*_{\tilde{\mu}}}{\sqrt{p_{\min}}}\sqrt{\bar{n}}T^{\frac{2}{3}} + L_1\frac{\sqrt{p_{\min}}}{\sqrt{\bar{n}}T^{\frac{2}{3}}}\mathbb{E}\left[\|G_{\tilde{\mu}}(x_0)\|^2\right] + L_1\frac{\sqrt{p_{\min}}}{\sqrt{\bar{n}}}T^{\frac{1}{3}}M + 2L_0^2\mu^2{L_1}^2(\bar{n}+3)^3T^{\frac{2}{3}}, \tag{34}$$

where $f^*_{\tilde{\mu}}$ is a lower bound on $\mathbb{E}[f_{\tilde{\mu}}(x)]$. The existance of such lower bound is due to (8), the definition of $\mathbb{E}[f_{\tilde{\mu}}(x)]$ and Assumption 1. The result in Theorem 1 follows by dividing both sides of the above inequality by T. □ □

The convergence rate shown above has the same order as that of applying the residual-feedback gradient estimator (4) to optimize a stochastic objective function as shown in [16]. This is because of this asynchronous scenario, the updates conducted by the other agents between two queries of a given agent introduce noise in the function evaluations from the perspective of this given agent. Furthermore, the bound on the non-stationarity of the solution in (34) increases as $p_{\min}$ becomes smaller. In practice, it means that the convergence of Algorithm 1 slows down if one of the agents is activated less frequently than others.

IV. SIMULATIONS

In this section, we demonstrate the effectiveness of the proposed asynchronous distributed zeroth-order optimization algorithm on a distributed feature learning example common in Internet of Things (IoT) applications. All the experiments are conducted using Python 3.8.5 on a 2017 iMac with 4.2GHz Quad-Core Intel Core i7 and 32GB 2400MHz DDR4.

Specifically, we consider the biomarker learning example described in [26], where a network of health monitoring edge devices collect heterogeneous raw input data $\{D_{i,j}\}_{i=1:N}$, e.g., different types of biosignals. Then, each device encodes its local raw data $D_{i,j}$ into a biomarker $d_{i,j}$ via a feature extraction function $\phi(D_{i,j}; x_i)$, e.g., a neural network with weights x_i, and sends it to a third-party entity that uses the collected biomarkers as predictors to learn a disease diagnosis for user j. The goal of the edge device i is to learn a better feature extraction function $\phi(\cdot; x_i)$ to help the third-party

entity to make better predictions. In practice, the prediction process at the third-party entity can be complicated and hard to model using an explicit function. Moreover, it may need to remain confidential. As a result, the edge device cannot obtain gradient information from this third-party entity. In the meantime, it is unreasonable to expect that the edge devices can update their feature extraction models synchronously or that they know the other edge devices' feature extraction models and parameters. Therefore, this problem presents an ideal case for the asynchronous distributed zeroth-order method proposed in this paper.

For simulation proposes, in this section, we assume that the third-party entity uses the logistic regression model

$$P(y_j; d_j) = 1/\big(1 + \exp\left(-y_i W^T d_j\right)\big), \tag{35}$$

where j represents the data point index, $y_j = \{1, -1\}$ and d_j denote the label and predictors for data point j, and W_T is a fixed classifier parameter. Specifically, let $d_j = [d_{1,j}, \ldots, d_{N,j}]^T$ represent the concatenated biomarker vector. The agents aim to collaboratively minimize the following loss function

$$f(\{x_i\}_{i=1:N}) = -\frac{1}{J}\sum_{j=1}^{J} \log\left(P(y_j; d_j(\cdot; \{x_i\}_{i=1:N})\right), \tag{36}$$

where J is the total number of data points.

Next, we apply the proposed Algorithm 1 to this distributed feature learning problem and compare its performance to an asynchronous extension of the centralized two-point gradient estimate (2) defined by

$$G_{\mu_i}(x_k) = \frac{f(x_k + \mu_i u_k) - f(x_{k-M})}{\mu_i} u_k, \tag{37}$$

where x_{k-M} is the most recent decision point agent i queries at iteration $k - M$.

Note that the convergence of the stochastic gradient descent update (11) using the asynchronous two-point estimator (37) has not been studied yet. We compare our proposed algorithm to the one with (37) simply to demonstrate the efficacy of our proposed approach. Specifically, we consider a network of 5 agents who collaboratively deal with $J = 20$ data samples. The feature extraction model $\phi(\cdot; x_i)$ at agent i is a single layer neural network with the input $D_{i,j} \in \mathbb{R}^{10}$ and a single output $d_{i,j} \in \mathbb{R}$. The activation function of the neural network is the sigmoid function. The weight of the neural network at agent i is denoted as x_i, which is initialized by sampling from a standard Gaussian distribution. We apply both the asynchronous residual-feedback gradient estimator (10) and the asynchronous two-point gradient estimator (37) to solve this problem. Specifically, for both gradient estimators, we run 10 trials. In addition, the smoothing parameter is $\mu = 0.1$, and the stepsizes α for gradient estimators (10) and (37) are selected as 0.5 and 0.5, respectively, so that they both achieve their fastest convergence speed during 10 trials of experiments. At each iteration, each agent has equal probability to be activated.

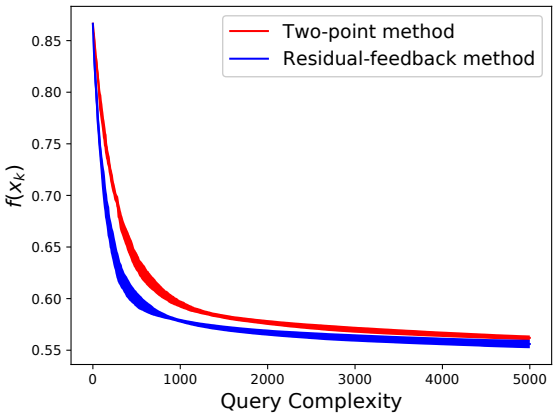

Fig. 1. Convergence results of the distributed feature learning problem. The red curve is obtained by applying the asynchronous two-point gradient estimator (37) and the blue curve is by the asynchronous residual-feedback estimator (10). The y axis denotes the value of the loss function (36) and the x axis represents the number of queries made in total by the team of agents. The shaded area around each curve represents the standard deviation of the function values over 10 trials.

The comparative performance results of using the two zeroth-order gradient estimators (10) and (37) are presented in Figure 1. We observe that during 10 trials, asynchronous learning with the residual-feedback gradient estimator (10) converges faster than asynchronous learning with the two-point gradient estimator (37). This is because the asynchronous residual-feedback gradient estimator (10) is subject to almost the same level of variance as the two-point gradient estimator (37), but can make twice the number of updates compared to the two-point gradient estimator (37) for the same number of queries. Note that we compare the two algorithms in terms of the number of queries rather than the number of updates, because the number of queries corresponds to the length of the global wall time required to run the algorithm.

V. CONCLUSIONS

In this paper, we proposed an asynchronous residual-feedback gradient estimator for distributed zeroth-order optimization, which estimates the gradient of the global cost function by querying the value of the function once at each time step. More importantly, only the local decision vector is needed for estimating the gradient and no communication among agents is required. We showed that the convergence rate of the proposed method matches the results for centralized residual-feedback methods when the function evaluation has noise. Numerical experiments on a distributed logistic regression problem are presented to show the effectiveness of the proposed method.

APPENDIX

Lemma 6. *Consider a sequence of non-negative real numbers* $\{V_k\}$ *with the following relations for all* $1 \leq k \leq T-1$, *provided with* $0 < \gamma + \beta < 1$,

$$V_k \leq \gamma\left(V_{k-1} + \beta V_{k-2} + \cdots + \beta^{k-1} V_0\right) + M,$$

where M is a contant, then we have

$$V_k \leq \gamma(\gamma+\beta)^{k-1}V_0 + \frac{1-\beta-\gamma(\gamma+\beta)^{k-1}}{1-(\gamma+\beta)}M.$$

In addition,

$$\sum_{k=0}^{T-1} V_k \leq \frac{1-\beta}{1-(\gamma+\beta)}V_0 + (T-1)\frac{1-\beta}{1-(\gamma+\beta)}M - \frac{\gamma}{(1-(\gamma+\beta))^2}M.$$

Proof. Fix some $k = K$. We have

$$\begin{aligned} V_K \leq & \gamma\left(V_{K-1} + \beta V_{K-2} + \cdots + \beta^{K-1}V_0\right) + M \\ \leq & \gamma(\gamma+\beta)\left(V_{K-2} + \cdots + \beta^{K-2}V_0\right) + \gamma M + M \end{aligned}$$

Repeat the above process, we obtain that

$$\begin{aligned} V_K \leq & \gamma(\gamma+\beta)^{K-2}\left(\gamma V_0 + M + \beta V_0\right) \\ & + \gamma\sum_{k=0}^{K-2}(\gamma+\beta)^k M + M \\ = & \gamma(\gamma+\beta)^{K-1}V_0 + \frac{1-\beta-\gamma(\gamma+\beta)^{K-1}}{1-(\gamma+\beta)}M, \end{aligned}$$

which completes the first part of the proof. Summing V_k from 0 to $T-1$, we obtain that

$$\begin{aligned} \sum_{k=0}^{T-1} V_k = & \sum_{k=1}^{T-1}\left(\gamma(\gamma+\beta)^{k-1}V_0 + \frac{1-\beta-\gamma(\gamma+\beta)^{k-1}}{1-(\gamma+\beta)}M\right) \\ & + V_0 \\ = & \frac{1-\beta}{1-(\gamma+\beta)}V_0 + (T-1)\frac{1-\beta}{1-(\gamma+\beta)}M \\ & - \frac{\gamma}{(1-(\gamma+\beta))^2}M, \end{aligned}$$

which complete the proof of the lemma. □ □

REFERENCES

[1] J. Konečnỳ, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," *arXiv preprint arXiv:1610.02527*, 2016.

[2] Y. Zhang and M. M. Zavlanos, "Distributed off-policy actor-critic reinforcement learning with policy consensus," in *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE, 2019, pp. 4674–4679.

[3] D. A. Schmidt, C. Shi, R. A. Berry, M. L. Honig, and W. Utschick, "Distributed resource allocation schemes," *IEEE Signal Processing Magazine*, vol. 26, no. 5, pp. 53–63, 2009.

[4] R. L. Raffard, C. J. Tomlin, and S. P. Boyd, "Distributed optimization for cooperative agents: Application to formation flight," in *2004 43rd IEEE Conference on Decision and Control (CDC)(IEEE Cat. No. 04CH37601)*, vol. 3. IEEE, 2004, pp. 2453–2459.

[5] S. Boyd, N. Parikh, and E. Chu, *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011.

[6] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[7] N. Chatzipanagiotis, D. Dentcheva, and M. M. Zavlanos, "An augmented lagrangian method for distributed optimization," *Mathematical Programming*, vol. 152, no. 1, pp. 405–434, 2015.

[8] Y. Zhang and M. M. Zavlanos, "A consensus-based distributed augmented lagrangian method," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 1763–1768.

[9] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization." *Journal of machine learning research*, vol. 13, no. 2, 2012.

[10] C.-C. Tu, P. Ting, P.-Y. Chen, S. Liu, H. Zhang, J. Yi, C.-J. Hsieh, and S.-M. Cheng, "Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 742–749.

[11] D. Bertsimas and A. J. Mersereau, "A learning approach for interactive marketing to a customer segment," *Operations Research*, vol. 55, no. 6, pp. 1120–1135, 2007.

[12] P. Mertikopoulos and Z. Zhou, "Learning in games with continuous action sets and unknown payoff functions," *Mathematical Programming*, vol. 173, no. 1, pp. 465–507, 2019.

[13] A. D. Flaxman, A. T. Kalai, and H. B. McMahan, "Online convex optimization in the bandit setting: gradient descent without a gradient," *arXiv preprint cs/0408007*, 2004.

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

[15] 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 Transactions on Information Theory*, vol. 61, no. 5, pp. 2788–2806, 2015.

[16] Y. Zhang, Y. Zhou, K. Ji, and M. M. Zavlanos, "Improving the convergence rate of one-point zeroth-order optimization using residual feedback," *arXiv preprint arXiv:2006.10820*, 2020.

[17] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE transactions on automatic control*, vol. 31, no. 9, pp. 803–812, 1986.

[18] A. Agarwal and J. C. Duchi, "Distributed delayed stochastic optimization," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*. IEEE, 2012, pp. 5451–5452.

[19] M. Zhong and C. G. Cassandras, "Asynchronous distributed optimization with minimal communication," in *2008 47th IEEE Conference on Decision and Control*. IEEE, 2008, pp. 363–368.

[20] H. Cai, Y. Lou, D. McKenzie, and W. Yin, "A zeroth-order block coordinate descent algorithm for huge-scale black-box optimization," *arXiv preprint arXiv:2102.10707*, 2021.

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

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

[23] Y. Zhang and M. M. Zavlanos, "Cooperative multi-agent reinforcement learning with partial observations," *arXiv preprint arXiv:2006.10822*, 2020.

[24] Y. Tang, Z. Ren, and N. Li, "Zeroth-order feedback optimization for cooperative multi-agent systems," in *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE, 2020, pp. 3649–3656.

[25] R. Durrett, *Essentials of stochastic processes*. Springer, 1999, vol. 1.

[26] B. Bent, K. Wang, E. Grzesiak, C. Jiang, Y. Qi, Y. Jiang, P. Cho, K. Zingler, F. I. Ogbeide, A. Zhao *et al.*, "The digital biomarker discovery pipeline: An open-source software platform for the development of digital biomarkers using mhealth and wearables data," *Journal of Clinical and Translational Science*, pp. 1–8, 2020.