2020 59th IEEE Conference on Decision and Control (CDC)
Jeju Island, Republic of Korea, December 14-18, 2020

Zeroth-Order Feedback Optimization for Cooperative Multi-Agent Systems

Yujie Tang, Zhaolin Ren, Na Li

***Abstract*— We consider a class of multi-agent optimization problems, where each agent is associated with an action vector and a local cost that depends on the joint actions of all agents, and the goal is to minimize the average of the local costs. Such problems arise in many control applications such as wind farm operation and mobile sensor coverage. In many of these applications, while we have access to (zeroth-order) information about function values, it can be difficult to obtain (first-order) gradient information. In this paper, we propose a zeroth-order feedback optimization (ZFO) algorithm based on two-point gradient estimators for the considered class of problems, and provide the convergence rate to a first-order stationary point for nonconvex problems. We complement our theoretical analysis with numerical simulations on a wind farm power maximization problem.**

I. INTRODUCTION

In this paper, we consider *cooperative multi-agent optimization* problems in the following form:

$$\min_{x^i\in\mathbb{R}^{d_i}} \ f(x^1,\ldots,x^n) = \frac{1}{n}\sum_{i=1}^n f_i(x^1,\ldots,x^n). \qquad (1)$$

Above, n is the number of agents, $x^i \in \mathbb{R}^{d_i}$ denotes the action vector of agent i, and $f_i : \mathbb{R}^{d_1}\times\cdots\times\mathbb{R}^{d_n} \to \mathbb{R}$ denotes the local cost function of agent i which is smooth but not necessarily convex. We note that the local cost f_i is a function of the joint action profile $x := (x^1,\ldots,x^n)$ for each i, indicating that the local cost value of agent i is affected not just by its own action x^i but also possibly the actions of all other agents. We also assume that each agent i can only control its own action vector x^i though the global system seeks to find an optimal joint action profile x. Therefore, our setting is distinct from the more commonly studied *global variable consensus optimization* (see [1] for a survey), where each agent maintains a local copy of the whole global decision variable, and is able to query information (e.g., function value, gradient) of the local cost evaluated at its own local copy without being directly affected by other agents.

Problem (1) and similar variants have appeared in many applications, such as wind farm power optimization [2], distributed routing control [3], and mobile sensor coverage [4]. In those applications, the goal is to seek local actions that optimize the global behavior of a system of closely interrelated agents. In many cases of these applications, each agent can only evaluate/observe the value of one part of the global cost, e.g., the function value of its own cost f_i and not (higher-order) derivatives thereof. This requires that the optimization procedure should be gradient-free, relying only on zeroth-order feedback of the local costs f_i. We refer to such optimization problems as *zeroth-order cooperative multi-agent optimization*.

*Yujie Tang, Zhaolin Ren, and Na Li are with the School of Engineering and Applied Sciences at Harvard University. Emails: (yujietang, nali)@seas.harvard.edu; zhaolinren@g.harvard.edu

While many multi-agent control problems can be formulated under the framework of zeroth-order cooperative multi-agent optimization, to the best of our knowledge, the design and analysis of effective algorithms for *general* zeroth-order cooperative multi-agent optimization has not been extensively studied in the literature (see Section I-B *Related Work* for more discussion on literature). This motivates our study of multi-agent optimization that leverages solely zeroth-order information.

A. Our Contributions

First, we propose a zeroth-order cooperative multi-agent optimization algorithm, which we call *Zeroth-order Feedback Optimization (ZFO)*. We consider the case where the individual cost functions f_i are smooth but nonconvex, and our goal is to find a joint action profile x close to a stationary point of the global cost f. The ZFO algorithm we develop depends on local computation and communication of two-point zeroth-order gradient estimators, such as the ones studied in [5], [6]. Specifically, at each iteration, each agent takes its own actions, collects zeroth-order information on the corresponding local costs, exchanges its most up-to-date zeroth-order information of all agents with its neighbors by a communication network, and forms a two-point zeroth-order gradient estimate to update its action vector. We note that the communication network could be subject to potential delays.

Second, we prove the convergence of our algorithm, and analyze its non-asymptotic convergence rate to a first-order stationary point when communication delays are bounded above. We focus on first-order stationarity as our setting involves nonconvex smooth objective functions. We establish a $O(1/\sqrt{T})$ convergence rate for the average squared norm of the gradients of the iterates, where T is the number of iterations. This dependence on T is the same as those results for related problems in the distributed nonconvex optimization literature, such as asynchronous parallel stochastic gradient descent [7] and distributed zeroth-order global consensus optimization [8]. We also analyze the dependence of the algorithm's sample complexity on the problem dimension, the network size, and the communication delays.

Third, empirically, we apply our algorithm to a simulated wind farm power maximization problem, and show that our algorithm can converge to an optimal solution for the given problem with satisfactory convergence behavior.

B. Related Work

a) Zeroth-order optimization: The absence of gradient information in our problem setting situates our work in the broader zeroth-order optimization literature. In the centralized setting, examples of zeroth-order optimization include estimating gradients using function values [5], [6], [9], [10], and direct-search methods that do not seek to approximate a gradient [11]. A survey of the zeroth-order literature can be found in [12]. There has also been some recent work utilizing zeroth-order optimization in a distributed setting [8], [13]–[16]. However, to the best of our knowledge, most of them focus on the distributed setting of global variable consensus optimization, rather than the cooperative multi-agent setting discussed in this paper.

b) Multi-agent optimization from game-theoretic perspective: The cooperative multi-agent optimization problem can also be interpreted as a cooperative game. Accordingly, there has been a line of work studying the problem from the lens of game theory. In particular, the problem studied in our work has also been studied in a series of papers utilizing game-theoretic control [2], [17], in the specific application of wind farm operation. However, [17] and [2] only provide results on asymptotic convergence, while we provide explicit convergence rates. In another related direction, [3] considers the problem of designing local objective functions so as to optimize global behavior in cooperative games. While our goal is similar, we consider the problem from an optimization rather than game-theoretic perspective, and assume only zeroth-order information of the local costs f_i is available.

c) Distributed optimization: Another research area closely related to our work is distributed optimization. While our setting is distinct from the extensively studied global variable consensus problem [18]–[24], we note that in both settings, agents need to collaborate so as to optimize the global objective. In addition, due to the local nature of communication, the agents will experience delays when receiving information from other (possibly distant) agents in the network. Therefore, our problem can also be viewed from the perspective of distributed optimization with delays [7], [25]–[28]. However, our work appears to be the first to study the effects of delays in a distributed zeroth-order setting.

Notation

Throughout this paper, we use $\|\cdot\|$ to denote the standard Euclidean norm. For any real-valued differentiable function $h(x) = h(x^1, \dots, x^n)$, we use $\nabla^i h(x)$ to denote the partial gradient of h with respect to x^i. The $p \times p$ identity matrix will be denoted by I_p. We use $\mathcal{N}(\mu, \Sigma)$ to denote the Gaussian distribution with mean μ and covariance matrix Σ.

II. Problem Formulation

Consider a group of n agents, where agent i is associated with an action vector $x^i \in \mathbb{R}^{d_i}$ for each $i = 1, \dots, n$. The joint action profile of the group of agents is then $x := (x^1, x^2, \dots, x^n) \in \mathbb{R}^d$, where $d = \sum_{i=1}^n d_i$. Each agent is also associated with a real-valued local cost function $f_i(x) = f_i(x^1, \dots, x^n)$ that depends on the joint action profile $x \in \mathbb{R}^d$. The goal of the agents is to cooperatively find the joint action profile that minimizes the average of the local costs, i.e., to solve the following problem

$$\min_{x \in \mathbb{R}^d} \; f(x) := \frac{1}{n} \sum_{i=1}^n f_i(x^1, \dots, x^n), \tag{2}$$

where $f(x)$ denotes the average costs among agents.

In solving this problem, it is natural that each agent needs to collect information on its own local cost and exchange information with other agents, as the local costs are affected by all agents' actions. We make two assumptions on how information is obtained and exchanged among agents. The first pertains to the type of information the agents can access, and the second to communication mechanism:

1) **Access to only zeroth-order information.** We assume that each agent i is only able to access (zeroth-order) function value information of its local cost f_i, and that derivatives of any order of f_i are not readily available. In obtaining the the function value, each agent i first updates its action vector x^i, which yields a new joint action profile $x = (x^1, \dots, x^n)$. Then, each agent i observes its corresponding local cost f_i evaluated at the updated $x = (x^1, \dots, x^n)$.

For simplicity, we assume that the function values observed by the agents are noiseless and accurate for most parts of this paper; we shall see in Section V-B that the proposed algorithm still works empirically when the function values are corrupted by independent additive noise.

2) **Localized communication.** We assume that the n agents are connected by a communication network. The topology of the communication network is represented by an undirected, connected graph $\mathcal{G} = (\{1, \dots, n\}, \mathcal{E})$, where the edges in $\mathcal{E}$ represent the bidirectional communication links. Each agent exchanges messages directly *only* with its neighbors in the communication network $\mathcal{G}$. As an example, if agent 1 is two hops away from agent 3 in $\mathcal{G}$, assuming no communication failure, it takes two communication rounds to transmit information from agent 1 to agent 3, and vice versa. We denote the distance (the length of the shortest path) between the pair of nodes (i, j) in the graph $\mathcal{G}$ by b_{ij}.

A. An Example: Wind Farm Power Maximization

Here we present an example abstracted from practical problems which fits the aforementioned formulation.

Consider a wind farm with n wind turbines. Each turbine i is associated with a local agent that is responsible for adjusting the turbine's axial induction factor denoted by a^i, which influences the amount of wind the turbine can harness. According to the Park model [29], a standard wake model studied in existing literature, when a wind turbine extracts energy out of the wind, it creates a wake downstream where the wind speed is reduced. As a result, the power generated by turbine i, denoted by P_i, depends not just on its own axial induction factor a^i but also on those of wind turbines upstream. Consequently, if we denote $a = (a^1, \ldots, a^n)$, then P_i is in general a function of the joint axial inductor factor profile a. The wind farm power maximization problem can now be posed as

$$\max_{a=(a^1,\ldots,a^n)} \frac{1}{n}\sum_{i=1}^{n} P_i(a^1,\ldots,a^n). \tag{3}$$

We assume the following mechanism of collecting and exchanging information among agents:

1) **Access to only zeroth-order information.** The agents do not have the computational capacity for numerical computation or simulation of P_i or its derivatives due to the highly complex aerodynamic interactions between turbines [2]. On the other hand, each agent i is able to measure the power P_i generated by its corresponding turbine via some measurement apparatus at any time.

2) **Localized communication.** The agents are connected by a bidirectional communication network, and each agent can only directly talk to its neighbors.

We refer to [2] for more details on the wind farm model and the power maximization problem.

III. Algorithm

A. Preliminaries on Zeroth-Order Gradient Estimators

In order to solve the problem (2) where only zeroth-order function value information can be obtained, we consider the following gradient estimator from zeroth-order optimization [5]:

$$\mathsf{G}_f(x;u,z) = \frac{f(x+uz) - f(x-uz)}{2u} z, \tag{4}$$

where u is a positive parameter called the *smoothing radius*, and z is sampled from the Gaussian distribution $\mathcal{N}(0, I_d)$. It can be shown (see Lemma 1) that when f is L-smooth, the bias of the estimator $\mathsf{G}_f(x;u,z)$ can be controlled by

$$\|\mathbb{E}_z[\mathsf{G}_f(x;u,z)] - \nabla f(x)\| \le uL\sqrt{d}.$$

In other words, $\mathsf{G}_f(x;u,z)$ can be viewed as a stochastic gradient with a nonzero bias that can be controlled by u. We can then plug this stochastic gradient into the gradient descent method, which leads to a zeroth-order optimization algorithm.

B. Our Proposed Algorithm

Our proposed algorithm is presented in Algorithm 1, which is based on the zeroth-order gradient estimator (4):

$$\mathsf{G}_f(x;u,z) = \frac{1}{n}\sum_{j=1}^{n} \frac{f_j(x+uz) - f_j(x-uz)}{2u} z$$

where $z \sim \mathcal{N}(0, I_d)$. We highlight that $\mathsf{G}_f(x;u,z)$ is a vector approximating the gradient $\nabla f(x) = \big(\nabla^1 f(x), \ldots, \nabla^n f(x)\big)$. One the one hand, each agent i only needs to estimate the partial gradient $\nabla^i f(x)$ to update its own action x^i. The estimation of the partial gradient is given by

$$\frac{1}{n}\sum_{j=1}^{n} \frac{f_j(x+uz) - f_j(x-uz)}{2u} z^i, \tag{7}$$

where $z^i \sim \mathcal{N}(0, I_{d_i})$ denotes the subvector of z corresponding to agent i's action vector. On the other hand, the computation of (7) requires agent i to collect the differences of function values $f_j(x+uz) - f_j(x-uz)$ of all agents j. While each agent can observe its own local cost, other agents' local cost information needs to be transmitted by the communication network and will suffer from delays. We therefore propose the following procedure for collecting and sharing necessary data among agents.

1) At time t, each agent i adjusts its actions to be $x^i(t) \pm$

Algorithm 1: Zeroth-order Feedback Optimization (ZFO) for cooperative multi-agent systems

Require: step size $\eta > 0$, smoothing radius $u > 0$, number of iterations T, initial point x_0

Initialize: $x(0) \leftarrow x_0, D_j^i(-1) = 0, \tau_j^i(-1) = -1$ for all $i, j = 1, \ldots, n$.

for $t = 0, \ldots, T-1$ **do**

Each agent i generates $z^i(t) \sim \mathcal{N}(0, I_{d_i})$.

Each agent i takes action $x^i(t) + uz^i(t)$.

Each agent i observes its local cost $\hat{f}_i^+(t)$.

Each agent i takes action $x^i(t) - uz^i(t)$.

Each agent i observes its local cost $\hat{f}_i^-(t)$.

Agent i computes

$$D_i^i(t) = \frac{\hat{f}_i^+(t) - \hat{f}_i^-(t)}{2u}, \qquad \tau_i^i(t) = t.$$

Agent i receives $\big(D_j^{k\to i}(t), \tau_j^{k\to i}(t)\big)_{j=1}^n$ from each neighbor $k : (k,i) \in \mathcal{E}$, and sets*

$$k_j^i(t) = \operatorname*{arg\,max}_{k:(k,i)\in\mathcal{E}} \tau_j^{k\to i}(t),$$

and

$$\tau_j^i(t) = \tau_j^{k_j^i(t)\to i}(t), \quad D_j^i(t) = D_j^{k_j^i(t)\to i}(t).$$

for each $j \ne i$.

Agent i sends $\big(D_j^i(t), \tau_j^i(t)\big)_{j=1}^n$ to its neighbors.

Agent i updates

$$G^i(t) = \frac{1}{n}\sum_{j:\tau_j^i(t)\ge 0} D_j^i(t)\, z^i(\tau_j^i(t)), \tag{5}$$

$$x^i(t+1) = x^i(t) - \eta\, G^i(t). \tag{6}$$

end

---

*In the situation where additional delay occurs in transmitting data from agent k to agent i, and agent i does not receive new data from agent k at time t, we let $\big(D_j^{k\to i}(t), \tau_j^{k\to i}(t)\big) = \big(D_j^i(t-1), \tau_j^i(t-1)\big)$.

$uz^i(t)$ and observes the corresponding local costs; see Lines 3-7 of Algorithm 1 where we denote $\hat{f}_i^{\pm}(t) = f_i(x(t) \pm uz(t))$. Agent i then computes

$$D_i^i(t) := \frac{f_i(x(t)+uz(t)) - f_i(x(t)-uz(t))}{2u},$$

and also records the time instant $\tau_i^i(t)$ at which $D_i^i(t)$ is computed (Line 8 in Algorithm 1). This pair of newly-generated data $(D_i^i(t), \tau_i^i(t))$ is going to be distributed via the communication network among agents.

2) At time t, agent i's most up-to-date information on f_j is recorded by a pair $\left(D_j^i(t), \tau_j^i(t)\right)$, where the quantity $D_j^i(t)$ records agent i's most up-to-date value of

$$\frac{f_j(x(\tau)+uz(\tau)) - f_j(x(\tau)-uz(\tau))}{2u},$$

and the quantity $\tau_j^i(t)$ records the time instant at which $D_j^i(t)$ was recorded by agent j. In other words,

$$\begin{aligned} &D_j^i(t) = D_j^j(\tau_j^i(t)) \\ &= \frac{f_j\big(x(\tau_j^i(t))+uz(\tau_j^i(t))\big) - f_j\big(x(\tau_j^i(t))-uz(\tau_j^i(t))\big)}{2u}. \end{aligned}$$

In order to update the pair $(D_j^i(t), \tau_j^i(t))$ for $j \neq i$ at time t, each agent i first receives data sent by its neighbors earlier, which we denote by $\left(D_j^{k\to i}(t), \tau_j^{k\to i}(t)\right)_{j=1}^n$ for each neighbor k. Then for each $j \neq i$, agent i finds the pair with the largest $\tau_j^{k\to i}(t)$, i.e., the pair with the most up-to-date information, and lets $(D_j^i(t), \tau_j^i(t))$ be equal to this pair (Line 9 in Algorithm 1) .

3) After all of the pairs $(D_j^i(t), \tau_j^i(t))$ have been updated, each agent i sends them to its neighbors in the network (Line 10 of Algorithm 1).

Here we further elaborate on this procedure and the communication delays therein: If each round of communication takes just one time step, and no additional delays occur during the communication (Lines 9 and 10) for all t, then agent i's received pair $\left(D_j^{k\to i}(t), \tau_j^{k\to i}(t)\right)$ is just $(D_j^k(t-1), \tau_j^k(t-1))$. Consequently $\tau_j^i(t) = t - b_{ij}$ for $t \geq b_{ij}$, as it takes exactly b_{ij} communication rounds to transmit data from agent j to agent i (recall that b_{ij} is the distance between i and j in $\mathcal{G}$). Correspondingly, $D_j^i(t) = D_j^j(t - b_{ij})$. On the other hand, when some additional delay occurs during communication, agent i may not receive new data from some neighbor k at some time step t. In this case we let $\left(D_j^{k\to i}(t), \tau_j^{k\to i}(t)\right) = \left(D_j^i(t-1), \tau_j^i(t-1)\right)$, i.e., agent i will just use old data. We shall see in Section IV that our algorithm works with performance guarantees when the additional delays during communication are bounded.

Lastly, each agent i calculates the subvector (7) but with delayed information. Specifically, one uses $D_j^i(t) = D_j^j(\tau_j^i(t))$ instead of $D_j^j(t)$, and correspondingly replaces $z^i(t)$ by $z^i(\tau_j^i(t))$ in (7); the resulting subvector is denoted by $G^i(t)$ in (5). A gradient descent step is then applied to obtain $x^i(t+1)$ (Line 11 of Algorithm 1).

Remark 1. Our study suggests that the gradient descent step (Line 11) of Algorithm 1 can be carried out by each agent in an asynchronous manner without sacrificing the convergence rate much. However, Lines 4 to 7 of Algorithm 1 require all agents to synchronize their changes of actions, so that $\hat{f}_i^+(t) - \hat{f}_i^-(t)$ gives the desired function value difference in the two-point zeroth-order gradient estimator (7). Whether these steps can also be made asynchronous without sacrificing the convergence rate is still under investigation.

IV. Convergence Analysis

In this section, we present our main results on the convergence rate of Algorithm 1. We first impose some assumptions on the objective functions.

Assumption 1. Each $f_j : \mathbb{R}^d \to \mathbb{R}$, $j = 1, \ldots, n$ is G-Lipschitz and L-smooth, i.e.,

$$\begin{aligned} |f_j(x) - f_j(y)| &\leq G\|x - y\|, \\ \|\nabla f_j(x) - \nabla f_j(y)\| &\leq L\|x - y\|, \end{aligned}$$

for all $x, y \in \mathbb{R}^d$. Moreover, $f^* := \inf_{x\in\mathbb{R}^d} f(x) > -\infty$.

Recalling that b_{ij} is the distance between i and j in the communication graph $\mathcal{G}$, we then have $t - \tau_j^i(t) \geq b_{ij}$, i.e., the delay $t - \tau_j^i(t)$ between agent j sending $(D_j^j(t), t)$ and agent i receiving $(D_j^i(t), \tau_j^i(t))$ is at least b_{ij}.

Assumption 2. There exists $\Delta \geq 0$ such that the delays are bounded above as $t - \tau_j^i(t) \leq b_{ij} + \Delta$ for any $t > 0$ and $i, j = 1, \ldots, n$.

We denote

$$\bar{b} := \sqrt{\frac{1}{n^2}\sum_{i=1}^n \sum_{j=1}^n (b_{ij} + \Delta)^2}, \qquad B := \max_{i,j} b_{ij} + \Delta.$$

Theorem 1 (Main result)**.** *Given the number of iterations $T > B$, let the step size and smoothing radius satisfy*

$$\eta = \frac{\alpha_\eta}{L\sqrt{\bar{b}\sqrt{n}d} \cdot \sqrt{T-B+1}}, \quad u^2 = \alpha_u^2 \frac{G^2}{L^2}\sqrt{\frac{\bar{b}\sqrt{n}}{(T-B+1)d}}$$

for some $\alpha_\eta \in (0, 1]$ and $\alpha_u > 0$. Then

$$\begin{aligned} &\frac{1}{T-B+1}\sum_{t=B}^T \mathbb{E}\big[\|\nabla f(x(t))\|^2\big] \\ &\leq G^2\left(\frac{6L(f(x_0) - f^*)}{5\alpha_\eta G^2} + 30\alpha_\eta + 2\alpha_u^2\right)\sqrt{\frac{\bar{b}\sqrt{n}d}{T - B + 1}} \\ &\quad + \frac{12\sqrt{3d}BG^2}{T - B + 1} \\ &= O\left(\sqrt{\frac{\bar{b}\sqrt{n}d}{T - B + 1}}\right). \end{aligned}$$

Corollary 1 (Sample complexity)**.** *Let $\epsilon > 0$ be arbitrary. The number of iterations T required for Algorithm 1 to achieve*

$$\frac{1}{T-B+1}\sum_{t=B}^T \mathbb{E}\big[\|\nabla f(x(t))\|^2\big] \leq \epsilon$$

is bounded by

$$T = O\left(\frac{\bar{b}\sqrt{n}d}{\epsilon^2}\right). \quad (8)$$

A proof sketch of Theorem 1 is given in Appendix I.

We now provide some discussions on the main theorem and its corollary.

1) **Ergodic convergence as the metric.** Since we consider smooth nonconvex objectives in (2), the commonly used convergence metrics in convex optimization (e.g., $f(x(t)) - f^*$ or $\|x(t) - x^*\|$) are not eligible unless further conditions are imposed. Instead, we consider ergodic convergence $\frac{1}{T-B+1}\sum_{t=B}^{T}\mathbb{E}\big[\|\nabla f(x(t))\|^2\big]$ that averages the (expected) squared norms of the gradients, which has been widely adopted in smooth nonconvex optimization [30], [31].

2) **Convergence rate.** The convergence rate in terms of the number of iterations T is $O(1/\sqrt{T})$. This rate is consistent with the centralized stochastic gradient descent without delay [30] and also the delayed stochastic gradient descent method [7].

3) **Dependence on problem dimension.** We can see from (8) that the sample complexity has an explicit linear dependence on d. This dependence is consistent with results in centralized and distributed zeroth-order optimization [5], [8] under the noiseless setting.

4) **Dependence on network size and delays.** Apart from d/ϵ^2, there is an additional factor $\bar{b}\sqrt{n}$ in the numerator of (8), which reflects the effects of the number of agents, the network structure and the communication delays. This dependence suggests that Algorithm 1 is able to scale reasonably with the size of the network. On the other hand, we are investigating whether this dependence can be further improved.

V. Simulations

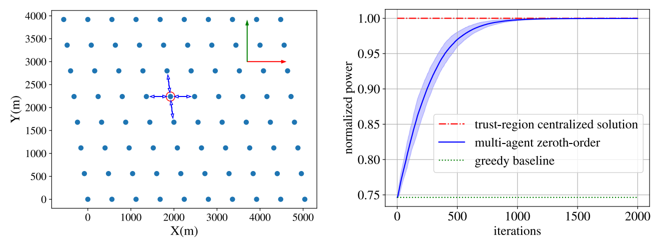
We demonstrate the performance of our ZFO algorithm on the power maximization problem in wind farms, which we have shown to be an example of the zeroth-order cooperative multi-agent optimization problem in Section II-A. For more details about the wind farm model we adopt, we refer the reader to Section II of [2].

A. Eighty-Turbine Example

We apply our ZFO algorithm to a setting with eighty turbines. We base our model on the Horns Rev wind farm in Denmark [32], whose layout is illustrated in Figure 1a (see also [2, Figure 5(a)]). In this model, the turbines (blue dots in Figure 1a) are placed in a parallelogram with 8 rows and 10 columns, and spaced 560 meters apart from each other in both X and Y directions. We let the wind blow in the positive X-direction. In addition, we assume the left, right, top and bottom turbines are neighbors in the communication network, so each turbine has up to 4 neighbors. As an example to demonstrate the typical connectivity in the communication graph, in Figure 1a, we draw an arrow between the turbine circled in red (fourth row from the top, fifth column from the left) and each of its four neighbors.

We introduce two benchmark action profiles for our simulation: One is the greedy solution maximizing each local objective given by $a_0 = (1/3, \ldots, 1/3)$ [2], and the other is an optimal action profile $a^* = (a^{1*}, \ldots, a^{n*})$ computed by a centralized trust-region algorithm. As a sanity check, the total generated power achieved by the greedy baseline a_0 is 74.6% of the optimal total power achieved by a^*, which is consistent with the empirical results in [2, Figure 5b].

For our ZFO algorithm, we normalize each $P_i(\cdot)$ by P^*/n where P^* is the total generated power under the optimal profile a^*. We pick the initial point to be a_0, and run our algorithm for $T = 2000$ iterations, with $\eta = 10^{-2}$ and $u = 0.075$. We repeat this for 50 trials, and assume exact access to function values.



(a) 2D Layout of turbines (b) Power trajectory plot

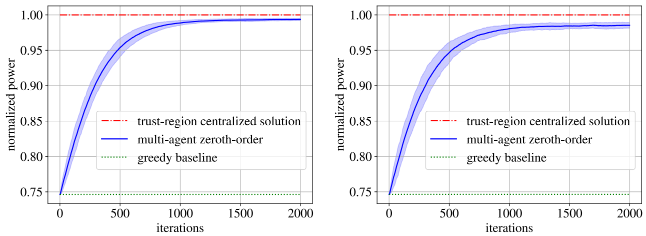
Fig. 1: Eighty-turbine simulation, exact function values, averaged over 50 trials.

In Figure 1, we plot the evolution of the total generated power normalized by P^* of our ZFO algorithm, as well as the greedy and optimal benchmarks. We use the dark blue line to indicate the average trajectory over 50 trials, and also include a light blue shaded band indicating a 2.0 standard deviation-sized confidence interval for the algorithm's power trajectory. As the plot in Figure 1b indicates, for our ZFO algorithm, the total generated power of the system converges to near the optimal value within about 1000 iterations.

We note that [2] also tested the performance of their algorithm numerically on the eighty-turbine Horns Rev wind farm. However, for this particular simulation, [2] used an algorithm which assumes that each agent has access to all other agents' costs, which is strictly stronger than ZFO's local communication assumption. Despite this, the convergence of ZFO in Figure 1b still seems faster than the corresponding convergence in [2] – the average trajectory of ZFO takes just under 500 iterations to be above 95% of the optimal power, while the algorithm in [2] takes about 1000 iterations (see [2, Figure 5(b)]).

B. The Setting with Noisy Zeroth-Order Information

In the preceding test case, we considered a deterministic setting where we enjoyed exact access to the function values for each agent. In this example, we adopt the same eighty-turbines wind farm model, but instead consider a noisy zeroth-order oracle: Given the joint action x, the value observed by agent i is $\hat{P}_i(x) := P_i(x) + \xi_i(x)$ for $\xi_i(x) \sim \mathcal{N}(0, \sigma^2)$; we also assume that noises added on different observations are independent. We test our algorithm with $\sigma = 0.1$ and $\sigma = 0.2$.



(a) $\sigma = 0.1$ (b) $\sigma = 0.2$

Fig. 2: Eighty-turbine simulation, noisy function values, averaged over 50 trials.

As we can see in Figures 2a and 2b, the variance of the trajectories in the noisy setting increases as we increase σ from 0.1 to 0.2. This is indicated by the light blue shaded bands illustrating the 2.0 standard deviation-sized confidence interval of the trajectories in both Figures 2a and 2b, and the fact that the band is thicker in Figure 2b ($\sigma = 0.2$) than in Figure 2a ($\sigma = 0.1$). In addition, comparing the two plots in Figure 2 for the noisy case to the plot in Figure 1b for the noiseless case, we observe that the speed of convergence degrades as the noise increases, and there seems to be a gap between the optimal value and the value the iterates converge to, which increases as the noise increases. Other than these observations, the average behavior across the exact and noisy settings seem broadly similar. While our current analysis holds only for deterministic two-point zeroth-order estimators, this simulation suggests that a similar result might hold in the noisy setting, making for interesting future work.

VI. Conclusion

In this paper, we propose a zeroth-order feedback optimization (ZFO) algorithm for cooperative multi-agent optimization. Theoretically, we prove the convergence of our algorithm for nonconvex cooperative multi-agent optimization, and provide explicit convergence rates. Numerically, we demonstrate that our algorithm is indeed convergent with appropriate parameter choices on a wind farm power maximization problem.

Some interesting future directions include 1) theoretical analysis for the case with noisy function evaluations, 2) how to handle constrained problems, 3) extension to asynchronous algorithms, 4) improvement of convergence rate and sample complexity.

Appendix I
Proof Sketch of Theorem 1

For notational simplicity, we let $D_j(t)$ denote $D_j^j(t)$ for $t \geq 0$, and let each $D_j(t) = 0$ and $z(t) = 0$ when $t < 0$. For each $t \geq 0$, let $G(t)$ be the d-dimensional vector that concatenates $G^1(t), \ldots, G^n(t)$ for each $i = 1, \ldots, n$. We see that each iteration of the algorithm can be equivalently written as

$$x(t+1) = x(t) - \eta\, G(t), \qquad t = 0, 1, 2, \ldots$$

Let $\mathcal{F}_t$ denote the σ-algebra generated by $x(\tau)$ for $\tau \leq t$ and all $\tau_j^i(s)$ for $1 \leq i, j \leq n$ and $0 \leq s \leq T$.

We introduce the "smoothed version" of f and f_j defined by $f^u(x) := \mathbb{E}_y[f(x+uy)]$ and $f_j^u(x) := \mathbb{E}_y[f_j(x+uy)]$ where $y \sim \mathcal{N}(0, I_d)$.

Lemma 1. *1) For each j, the function $f_j^u : \mathbb{R}^d \to \mathbb{R}$ is G-Lipschitz and L-smooth, and for each $x \in \mathbb{R}^d$,*

$$\mathbb{E}_z\left[\frac{f_j(x+uz) - f_j(x-uz)}{2u} z\right] = \nabla f_j^u(x), \quad z \sim \mathcal{N}(0, I_d).$$

2) For any $x \in \mathbb{R}^d$, we have

$$\|\nabla f(x) - \nabla f^u(x)\| \leq uL\sqrt{d}.$$

The first part of Lemma 1 can be found in [5], and a proof of the second part follows similarly as [33, Lemma 6(b)]. Then by appealing to concentration inequalities for standard Gaussian distribution and following the derivations in [6, Lemma 10], we get the following lemma.

Lemma 2. *Let $z \sim \mathcal{N}(0, I_d)$, and let $h : \mathbb{R}^d \to \mathbb{R}$ be G-Lipschitz. Then*

$$\mathbb{E}_z\left[\left\|\frac{h(x+uz) - h(x-uz)}{2u} z_i\right\|^2\right] \leq 12G^2,$$

for any $i = 1, \ldots, d$, where z_i denotes the i'th entry of z.

Lemma 2 helps to bound terms related to the second moment of $G^i(t)$.

Lemma 3. *For any $t \geq 0$, we have*

$$\mathbb{E}\left[\left\|D_j(\tau_j^i(t)) z^i(\tau_j^i(t))\right\|^2\right] \leq 12G^2 d_i,$$
$$\mathbb{E}\left[\|G(t)\|^2\right] \leq 12G^2 d.$$

Proof. The first inequality is a consequence of Lemma 2. For the second inequality, we have

$$\begin{aligned}
\mathbb{E}\left[\|G(t)\|^2\right] &= \sum_{i=1}^n \mathbb{E}\left[\left\|\frac{1}{n}\sum_{j=1}^n D_j(\tau_j^i(t)) z^i(\tau_j^i(t))\right\|^2\right] \\
&\leq \frac{1}{n}\sum_{i=1}^n \sum_{j=1}^n \mathbb{E}\left[\left\|D_j(\tau_j^i(t)) z^i(\tau_j^i(t))\right\|^2\right] \\
&\leq \sum_{i=1}^n 12G^2 d_i \leq 12G^2 d.
\end{aligned}$$

□

The following lemma will be used to quantify the effect of delays on Algorithm 1.

Lemma 4. *Let $i, j \in \mathcal{N}$ be arbitrary. Then for any $t \geq b_{ij} + \Delta$,*

$$\begin{aligned}
&\mathbb{E}\left[\left\|\nabla^i f_j^u\big(x(t)\big) - \nabla^i f_j^u\big(x(\tau_j^i(t))\big)\right\|^2\right] \\
&\leq 12G^2\eta^2 L^2 (b_{ij} + \Delta)^2 d,
\end{aligned}$$

and

$$\begin{aligned}
&\mathbb{E}\left[\left\|\nabla^i f\big(x(t)\big) - \nabla^i f\big(x(\tau_j^i(t))\big)\right\|^2\right] \\
&\leq 12G^2\eta^2 L^2 (b_{ij} + \Delta)^2 d.
\end{aligned}$$

Proof. Since f_j^u is L-Lipschitz, we have

$$\begin{aligned}
&\mathbb{E}\Big[\big\|\nabla^i f_j^u\big(x(t)\big) - \nabla^i f_j^u\big(x(\tau_j^i(t))\big)\big\|^2\Big] \\
&\leq L^2\,\mathbb{E}\Big[\big\|x(t) - x(\tau_j^i(t))\big\|^2\Big] \\
&\leq L^2\,\mathbb{E}\bigg[\Big(\textstyle\sum_{\tau=-b_{ij}-\Delta}^{-1}\|\eta\, G(t+\tau)\|\Big)^2\bigg] \\
&\leq \eta^2 L^2 (b_{ij}+\Delta)\sum_{\tau=-b_{ij}-\Delta}^{-1}\mathbb{E}\big[\|G(t+\tau)\|^2\big] \\
&\leq 12G^2\eta^2L^2(b_{ij}+\Delta)^2 d.
\end{aligned}$$

The second inequality follows similarly. □

Lemma 5. *For any $t \geq B$, we have*

$$\begin{aligned}
&\mathbb{E}\bigg[\frac{-1}{n}\sum_{i,j}\big\langle\nabla^i f\big(x(t)\big) - \nabla^i f\big(x(\tau_j^i(t))\big), D_j(\tau_j^i(t))z^i(\tau_j^i(t))\big\rangle\bigg] \\
&\leq 12G^2\eta L\bar{b}\sqrt{n}d,
\end{aligned}$$

where $\langle\cdot,\cdot\rangle$ denotes the standard Euclidean inner product.

Proof. We have

$$\begin{aligned}
&\mathbb{E}\bigg[\frac{-1}{n}\sum_{i,j}\big\langle\nabla^i f\big(x(t)\big) - \nabla^i f\big(x(\tau_j^i(t))\big), D_j(\tau_j^i(t))z^i(\tau_j^i(t))\big\rangle\bigg] \\
&\leq \frac{1}{2n}\cdot\frac{1}{\eta L\bar{b}\sqrt{n}}\sum_{i,j}\mathbb{E}\Big[\big\|\nabla^i f(x(t)) - \nabla^i f\big(x(\tau_j^i(t))\big)\big\|^2\Big] \\
&\quad + \frac{1}{2n}\cdot\eta L\bar{b}\sqrt{n}\sum_{i,j}\mathbb{E}\left[\|D_j(\tau_j^i(t))z^i(\tau_j^i(t))\|^2\right] \\
&\leq \frac{12G^2\eta L d}{2n^{3/2}\bar{b}}\sum_{i,j}(b_{ij}+\Delta)^2 + \frac{12G^2\eta L\bar{b}\sqrt{n}}{2}\cdot\frac{1}{n}\sum_{j=1}^{n}\sum_{i=1}^{n} d_i \\
&= 12G^2\eta L\bar{b}\sqrt{n}d.
\end{aligned}$$

Here in the first inequality we used the fact that $2\langle u,v\rangle \leq \|u\|^2/\epsilon + \epsilon\|v\|^2$ for any $\epsilon > 0$ and any vectors u, v, and in the third inequality we used Lemmas 2 and 4. □

Lemma 6. *For any $t \geq B$, we have*

$$\begin{aligned}
&\mathbb{E}\bigg[-\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n}\big\langle\nabla^i f\big(x(\tau_j^i(t))\big), D_j(\tau_j^i(t))z^i(\tau_j^i(t))\big\rangle\bigg] \\
&\leq -\frac{5}{6}\mathbb{E}\big[\|\nabla f(x(t))\|^2\big] + 4\sqrt{3}G^2\eta L\bar{b}\sqrt{n}d + \frac{3}{2}u^2L^2d.
\end{aligned}$$

Proof sketch. By Lemma 1, for $t - b_{ij} - \Delta \leq \tau \leq t - b_{ij}$,

$$\begin{aligned}
&\mathbb{E}\Big[-\big\langle\nabla^i f\big(x(\tau)\big), D_j(\tau)z^i(\tau)\big\rangle\cdot\mathbf{1}_{\tau_j^i(t)=\tau}\Big|\mathcal{F}_\tau\Big] \\
&= -\big\langle\nabla^i f(x(\tau)), \nabla^i f_j^u(x(\tau))\big\rangle\cdot\mathbf{1}_{\tau_j^i(t)=\tau},
\end{aligned}$$

where $\mathbf{1}$ denotes the indicator function of an event. Therefore

$$\begin{aligned}
&\mathbb{E}\big[-\big\langle\nabla^i f\big(x(\tau_j^i(t))\big), D_j(\tau_j^i(t))z^i(\tau_j^i(t))\big\rangle\big] \\
&= \sum_\tau\mathbb{E}\Big[\mathbb{E}\Big[-\big\langle\nabla^i f\big(x(\tau)\big), D_j(\tau)z^i(\tau)\big\rangle\,\mathbf{1}_{\tau_j^i(t)=\tau}|\mathcal{F}_\tau\Big]\Big] \\
&= \mathbb{E}\bigg[-\sum_\tau\big\langle\nabla^i f(x(\tau)), \nabla^i f_j^u\big(x(\tau)\big)\big\rangle\cdot\mathbf{1}_{\tau_j^i(t)=\tau}\bigg] \\
&= \mathbb{E}\big[-\big\langle\nabla^i f\big(x(\tau_j^i(t))\big), \nabla^i f_j^u\big(x(\tau_j^i(t))\big)\big\rangle\big].
\end{aligned}$$

We then notice that

$$\begin{aligned}
&-\frac{1}{n}\sum_{i,j}\big\langle\nabla^i f\big(x(\tau_j^i(t))\big), \nabla^i f_j^u\big(x(\tau_j^i(t))\big)\big\rangle \\
&= -\frac{1}{n}\sum_{i,j}\big\langle\nabla^i f\big(x(\tau_j^i(t))\big) - \nabla^i f\big(x(t)\big), \nabla^i f_j^u\big(x(\tau_j^i(t))\big)\big\rangle \\
&\quad -\frac{1}{n}\sum_{i,j}\big\langle\nabla^i f\big(x(t)\big), \nabla^i f_j^u\big(x(\tau_j^i(t))\big) - \nabla^i f_j^u\big(x(t)\big)\big\rangle \\
&\quad - \langle\nabla f(x(t)), \nabla f^u(x(t)) - \nabla f(x(t))\rangle - \|\nabla f(x(t))\|^2.
\end{aligned}$$

It can be shown that the expected value of each term on the right-hand side can be individually bounded as

$$\begin{aligned}
&\mathbb{E}\bigg[-\frac{1}{n}\sum_{i,j}\big\langle\nabla^i f\big(x(\tau_j^i(t))\big) - \nabla^i f\big(x(t)\big), \nabla^i f_j^u\big(x(\tau_j^i(t))\big)\big\rangle\bigg] \\
&\leq 2\sqrt{3}G^2\eta L\bar{b}\sqrt{n}d, \\
&\mathbb{E}\bigg[-\frac{1}{n}\sum_{i,j}\big\langle\nabla^i f\big(x(t)\big), \nabla^i f_j^u\big(x(\tau_j^i(t))\big) - \nabla^i f_j^u\big(x(t)\big)\big\rangle\bigg] \\
&\leq 2\sqrt{3}G^2\eta L\bar{b}\sqrt{n}d,
\end{aligned}$$

and

$$\begin{aligned}
&\mathbb{E}\left[-\langle\nabla f(x(t)), \nabla f^u(x(t)) - \nabla f(x(t))\rangle\right] \\
&\leq \frac{1}{6}\mathbb{E}\big[\|\nabla f(x(t))\|^2\big] + \frac{3}{2}u^2L^2d,
\end{aligned}$$

where we skip the details due to space limit, but point out that their derivations follow mostly from

$$\mathbb{E}[\langle u,v\rangle] \leq \sqrt{\mathbb{E}[\|u\|^2]}\sqrt{\mathbb{E}[\|v\|^2]} \leq \mathbb{E}\big[\|u\|^2/(2\epsilon) + 2\epsilon\|v\|^2\big]$$

for any $\epsilon > 0$ and the previous lemmas. The lemma's conclusion then follows by adding these inequalities together. □

We are now ready to prove Theorem 1. By the L-smoothness of f, we see that

$$\begin{aligned}
f(x(t+1)) &\leq f(x(t)) - \eta\langle\nabla f(x(t)), G(t)\rangle + \frac{\eta^2 L}{2}\|G(t)\|^2 \\
&= f(x(t)) - \sum_{i=1}^{n}\eta\langle\nabla^i f(x(t)), G^i(t)\rangle + \frac{\eta^2 L}{2}\|G(t)\|^2.
\end{aligned}$$

We have

$$\begin{aligned}
&-\sum_{i=1}^{n}\langle\nabla^i f(x(t)), G^i(t)\rangle \\
&= -\frac{1}{n}\sum_{i,j}\big\langle\nabla^i f(x(t)) - \nabla^i f\big(x(\tau_j^i(t))\big), D_j(\tau_j^i(t))\,z^i(\tau_j^i(t))\big\rangle \\
&\quad -\frac{1}{n}\sum_{i,j}\langle\nabla^i f\big(x(\tau_j^i(t))\big), D_j(\tau_j^i(t))\,z^i(\tau_j^i(t))\rangle,
\end{aligned}$$

and by Lemmas 5 and 6, we get

$$\begin{aligned}
&\mathbb{E}\bigg[-\sum_{i=1}^{n}\langle\nabla^i f(x(t)), G^i(t)\rangle\bigg] \\
&\leq -\frac{5}{6}\mathbb{E}\big[\|\nabla f(x(t))\|^2\big] + 19G^2\eta L\bar{b}\sqrt{n}d + \frac{3}{2}u^2L^2d,
\end{aligned}$$

and together with the bound in Lemma 3, we get

$$\mathbb{E}[f(x(t+1))] \leq \mathbb{E}[f(x(t))] - \frac{5\eta}{6}\mathbb{E}\left[\|\nabla f(x(t))\|^2\right] + 25G^2\eta^2 L\bar{b}\sqrt{n}d + \frac{3}{2}\eta u^2 L^2 d.$$

By taking the telescoping sum, we get

$$\frac{1}{T-B+1}\sum_{t=B}^{T}\mathbb{E}\left[\|\nabla f(x(t))\|^2\right] \leq \frac{6\mathbb{E}[f(x(B)) - f^*]}{5\eta(T-B+1)} + 30G^2\eta L\bar{b}\sqrt{n}d + 2u^2L^2d.$$

For $\mathbb{E}[f(x(B))]$, we have

$$\mathbb{E}[f(x(B))] - f(x_0) \leq G\mathbb{E}\|x(B) - x(0)\|] \leq \eta G\sum_{t=0}^{B-1}\mathbb{E}[\|G(t)\|] \leq \eta GB\sqrt{12G^2d}.$$

Therefore

$$\frac{1}{T-B+1}\sum_{t=B}^{T}\mathbb{E}\left[\|\nabla f(x(t))\|^2\right] \leq \frac{6\left(f(x_0) - f^*\right)}{5\eta(T-B+1)} + 30G^2\eta L\bar{b}\sqrt{n}d + 2u^2L^2d + \frac{12\sqrt{3d}G^2B}{5(T-B+1)}.$$

By plugging in the given step size η and smoothing radius u, we get the desired convergence rate.

References

[1] A. Nedić and A. Ozdaglar, "Cooperative distributed multi-agent optimization," in *Convex Optimization in Signal Processing and Communications*, D. P. Palomar and Y. C. Eldar, Eds. Cambridge University Press, 2010, pp. 340–386.

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

[3] N. Li and J. R. Marden, "Designing games for distributed optimization," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 2, pp. 230–242, 2013.

[4] J. Cortés, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.

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

[6] O. Shamir, "An optimal algorithm for bandit and zero-order convex optimization with two-point feedback," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 1703–1713, 2017.

[7] X. Lian, Y. Huang, Y. Li, and J. Liu, "Asynchronous parallel stochastic gradient for nonconvex optimization," in *Advances in Neural Information Processing Systems*, 2015, pp. 2737–2745.

[8] Y. Tang and N. Li, "Distributed zero-order algorithms for nonconvex multi-agent optimization," in *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2019, pp. 781–786.

[9] 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.

[10] A. D. Flaxman, A. T. Kalai, and H. B. McMahan, "Online convex optimization in the bandit setting: gradient descent without a gradient," in *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, 2005, pp. 385–394.

[11] V. Torczon, "On the convergence of pattern search algorithms," *SIAM Journal on Optimization*, vol. 7, no. 1, pp. 1–25, 1997.

[12] J. Larson, M. Menickelly, and S. M. Wild, "Derivative-free optimization methods," *Acta Numerica*, vol. 28, pp. 287–404, 2019.

[13] 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.

[14] Y. Li, Y. Tang, R. Zhang, and N. Li, "Distributed reinforcement learning for decentralized linear quadratic control: A derivative-free policy optimization approach," *arXiv preprint arXiv:1912.09135*, 2019.

[15] A. K. Sahu, D. Jakovetic, D. Bajovic, and S. Kar, "Distributed zeroth order optimization over random networks: A Kiefer-Wolfowitz stochastic approximation approach," in *Proceedings of the 57th IEEE Conference on Decision and Control*, 2018, pp. 4951–4958.

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

[17] J. Marden, S. Ruben, and L. Pao, "Surveying game theoretic approaches for wind farm optimization," in *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, 2012. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2012-1154

[18] 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.

[19] I. Lobel and A. Ozdaglar, "Distributed subgradient methods for convex optimization over random networks," *IEEE Transactions on Automatic Control*, vol. 56, no. 6, pp. 1291–1306, 2010.

[20] T.-H. Chang, M. Hong, and X. Wang, "Multi-agent distributed optimization via inexact consensus ADMM," *IEEE Transactions on Signal Processing*, vol. 63, no. 2, pp. 482–497, 2014.

[21] 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.

[22] G. Qu and N. Li, "Harnessing smoothness to accelerate distributed optimization," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 3, pp. 1245–1260, 2017.

[23] T. Tatarenko and B. Touri, "Non-convex distributed optimization," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3744–3757, 2017.

[24] S. Pu and A. Nedić, "A distributed stochastic gradient tracking method," in *Proceedings of the 57th IEEE Conference on Decision and Control*, 2018, pp. 963–968.

[25] A. Nedić, "Asynchronous broadcast-based convex optimization over a network," *IEEE Transactions on Automatic Control*, vol. 56, no. 6, pp. 1337–1351, 2010.

[26] A. Agarwal and J. C. Duchi, "Distributed delayed stochastic optimization," in *Advances in Neural Information Processing Systems*, 2011, pp. 873–881.

[27] R. Zhang and J. Kwok, "Asynchronous distributed ADMM for consensus optimization," in *Proceedings of the 31st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 32, no. 2, 2014, pp. 1701–1709.

[28] J. Liu and S. J. Wright, "Asynchronous stochastic coordinate descent: Parallelism and convergence properties," *SIAM Journal on Optimization*, vol. 25, no. 1, pp. 351–376, 2015.

[29] A. K. Scholbrock, "Optimizing wind farm control strategies to minimize wake loss effects," Master's thesis, University of Colorado, Boulder, 2011.

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

[31] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu, "Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent," in *Advances in Neural Information Processing Systems*, 2017, pp. 5330–5340.

[32] R. J. Barthelmie, S. T. Frandsen, K. Hansen, J. G. Schepers, K. Rados, W. Schlez, A. Neubert, L. E. Jensen, and S. Neckelmann, "Modelling the impact of wakes on power output at Nysted and Horns Rev," in *European Wind Energy Conference*, 2009.

[33] D. Malik, A. Pananjady, K. Bhatia, K. Khamaru, P. Bartlett, and M. Wainwright, "Derivative-free methods for policy optimization: Guarantees for linear quadratic systems," in *Proceedings of Machine Learning Research*, vol. 89, 2019, pp. 2916–2925.