Distributed Stochastic Optimization in Networks With Low Informational Exchange

Wenjie Li, Member, IEEE, and Mohamad Assaad, Senior Member, IEEE

Abstract—We consider a distributed stochastic optimization problem in networks with finite number of nodes. Each node adjusts its action to optimize the global utility of the network, which is defined as the sum of local utilities of all nodes. While Gradient descent method is a common technique to solve such optimization problem, the computation of the gradient may require much information exchange. In this paper, we consider that each node can only have a noisy numerical observation of its local utility, of which the closed-form expression is not available. This assumption is quite realistic, especially when the system is either too complex or constantly changing. Nodes may exchange partially the observation of their local utilities to estimate the global utility at each timeslot. We propose a distributed algorithm based on stochastic perturbation, under the assumption that each node has only part of the local utilities of the other nodes. We use stochastic approximation tools to prove that our algorithm converges almost surely to the optimum, given that the objective function is smooth and strictly concave. The convergence rate is also derived, under the additional assumption of strongly concave objective function. It is shown that the convergence rate scales as $O\left(\frac{1}{K}\right)$ after a sufficient number of iterations $K > K_0$, which is the optimal rate order in terms of $K$ for our problem. Although the proposed algorithm can be applied to general optimization problems, we perform simulations for a typical power control problem in wireless networks and present numerical results to corroborate our claims.

Index Terms—Optimization, derivative-free optimization, stochastic approximation, convergence analysis, distributed algorithms.

I. INTRODUCTION

DISTRIBUTED optimization is a fundamental problem in networks, which helps to improve the performance of the system by maximizing some predefined objective function. Significant work has been done to solve such optimization problems in various applications. For example, power control [2]–[4] and beamforming allocation [5]–[8] problems have been widely studied in wireless communications systems. In such problems, transmitters need to control their transmission power or beamforming in a smart manner, in order to maximize some performance metric such as throughput or energy efficiency. In medium access control problem [9], users set their individual channel access probability in such a way to maximize their benefit. Distributed optimization is also of interest in wireless sensor networks, where sensor nodes collect information to serve a fusion center. A typical example is when each sensor node has to decide independently the quality of its report in such a way to maximize the average quality of information gathered by the fusion center, subject to a maximum power constraint consumed by the sensors [10], [11].

This paper considers an optimization problem in a distributed network where each node adjusts locally its own action, with the objective to maximize the global utility of the system. The global utility is the sum of the local utilities of all nodes in the network and is perturbed by a stochastic process, e.g., wireless channels. Gradient descent method is the most common technique to deal with optimization problems. However, in many scenarios in practice, the gradient computation may require too much information exchange between the nodes (an example is provided in Section VII-A). Furthermore, there are other contexts also where either the utility function of each node does not have a closed form expression or the expression of the utility function is very complex. In such cases, the computation of the derivatives is very complicated and may be not possible, which prevents from using the gradient-based method.

In this paper, we are interested in such contexts where the gradient cannot be derived by each node. In particular, we consider that each node has a noisy numerical observation of its utility function. This assumption is quite realistic when the system is complex and time-varying. In order to make each node independently evaluate the global utility of the network, the nodes may have to exchange their local utilities with some (not all) other nodes. In particular, we consider that a node can receive the local utility of another node with a given probability. This assumption of partial exchange of local utilities is made mainly to limit the signaling overhead in the network, and can be due to the network topology in some cases as well. As a result, each node has to approximate the global utility with only incomplete information of other local utilities. More interestingly, different nodes may have different incomplete information of local utilities, which
makes developing a distributed optimization method more challenging.

In summary, our problem is quite challenging due to the following reasons: i) each node has only a numerical observation of its local utility at each time; ii) each node may have incomplete information of the global utility of the network and this incomplete information may change from one node to another; iii) the action of each node has an impact on the utilities of the other nodes in the network; iv) the utility of each node is also influenced by some stochastic process (e.g., time varying channels in wireless networks) and the objective function is the average global utility.

Since only numerical observation of utilities can be obtained, our problem falls into the framework of derivative-free stochastic optimization (DFSO), and it can be also named as zero-order stochastic optimization. It has been well understood that there is a fundamental performance gap between DFSO and gradient-based optimization. Several lower bounds of the convergence rate have been derived in [12]–[14] to state that, without the knowledge of gradient, the optimization error cannot be better than $O(K^{-0.5})$ after $K$ iterations for strongly concave objective functions. This result holds true for smooth and strongly concave functions [13] as well. Note that when the gradient is available, the convergence rate can be as fast as $O(K^{-1})$ given that the objective function is strongly concave. Although DFSO has attracted much attention in learning community in recent years, the existing algorithms are usually centralized and not easy to be decentralized. For example, the algorithms in [15]–[17] contain the operations of vectors and matrices that require a control center to handle. In more details, even if each node has a perfect knowledge of the utilities of all other nodes, these existing DFSO algorithms will still require a control center and cannot be applied in a decentralized way, since the action of each node is based on manipulations of vectors and matrices that are common to all nodes. More discussions on the exiting DFSO methods will be provided in Section II. In our setting, each node controls only its local variable and may receive feedback from part of the other nodes. We develop a new DFSO algorithm to optimize the average global utility, which is the sum of all local utilities. Our algorithm is named Distributed Optimization algorithm using Stochastic Perturbation (DOSP). In our algorithm, each node will first estimate or approximate the global utility function at each time. Obviously two different nodes may have two different instantaneous estimates of the global utility (i.e. objective function). Each node will then independently updates its local action at each time (or iteration), according to its own numerical approximation/estimation of the current global utility. The main idea of our algorithm is to attempt to estimate the gradient by: i) using only a single approximated value of the objective function by each node at each iteration and, ii) introducing an appropriate stochastic perturbation in the action/decision in such a way that each node can make its action independently from other nodes. It is worth mentioning that our algorithm results in a biased gradient estimation as proved in this paper. By using stochastic approximation tools, we prove in this paper that our distributed DFSO algorithm converges almost surely to the global optimum, given that the objective function is smooth and strictly concave. An important step in the convergence proof is to derive an upper bound on the gradient estimation bias achieved by our algorithm, and then to show that this bias vanishes when the number of iterations tends to infinity. In addition, we analyze the convergence rate of our algorithm by: i) first deriving an upper bound of the average divergence between the action decided by our algorithm and the optimal one at an arbitrary iteration $k$, and ii) deeply investigating the impact of the parameters introduced by our algorithm on this bound. Interestingly, under the assumption that the objective function is smooth and strongly concave, we prove that the convergence rate of our algorithm is of order $O(K^{-0.5})$ after sufficient number of iterations $K > K_0$, which coincides with the lower bounds in [12]–[14]. This implies that, although we consider a challenging distributed setting, the achieved convergence rate (i.e. $O(K^{-0.5})$) of our algorithm is optimal for smooth and strongly concave functions.

Some preliminary results of our work have been presented in [1]. This extended version provides the complete proof of all the results. Moreover, we assume in this work that the action of each node belongs to a predefined compact set and provide an extensive analysis of the convergence rate, which is not considered at all in [1]. It is worth mentioning that the derivation of the convergence rate is much more challenging than that of classical stochastic approximation. In fact, our derivative-free approach leads to a biased estimation of the gradient, and therefore we have to analyze an additional term associated to the estimation bias of gradient.

The rest of the paper is organized as follows. Section II discusses some related work and highlights our main contribution. Section III describes the system model as well as some basic assumptions. Section IV presents our distributed optimization algorithm using stochastic perturbation (DOSP). Section V discusses the almost sure convergence of our DOSP algorithm. Section VI focuses on the analysis of the convergence rate of DOSP. Section VII shows some numerical results as well as a comparison with an alternative algorithm and Section VIII concludes this paper.

II. RELATED WORK

Most of the prior work in the area of optimization consider that the objective function has a well known and simple closed form expression. Under this assumption, the optimization problem can be solved using gradient ascent or descent method [18]. This method can achieve either a local optimum or a global optimum of the problem in some special cases (e.g. concavity of the utility, etc.). Several variants of the gradient-based method have been proposed in the literature. For example, a distributed asynchronous stochastic gradient optimization algorithm is presented in [19]. Incremental subgradient methods for non-differentiable optimization are discussed in [20]. Interested readers are referred to [21] for a survey on incremental gradient, sub-gradient, and proximal methods for convex optimization. The use of gradient-based method supposes in advance that the gradient can be computed.
or is available at each node, which is not always possible as this would require too much information exchanges. In our case, the computation of the gradient is not possible at each node since only limited control information can be exchanged in the network. Our goal is to develop an algorithm that requires only the knowledge of a numerical observation of the utility function. The resulting algorithm should be also distributed.

Distributed optimization has also been studied in the literature using game theoretic tools. However, most of the existing work assume that a closed form expression of the payoff is available. One can refer to [22], [23] and the references therein for more details, while we do not consider a non-cooperative game in this paper.

Stochastic approximation (SA) [24], [25] is an efficient method to solve optimization problems in noisy environment. Typically, the action is updated as follows

$$a_{k+1} = a_k + \beta_k \hat{g}_k.$$  
(1)

where $\hat{g}_k$ represents an unbiased estimation of the gradient $g_k$ with $\mathbb{E}[\hat{g}_k] = g_k$. An important assumption is that the estimation error $\epsilon_k = \hat{g}_k - g_k$ is seen as a zero-mean random vector with finite variance, e.g. see [26]–[29]. If the step-size $\beta_k$ is properly chosen, then $a_k$ can tend to its optimum point asymptotically. The challenge of our work is how to propose such an estimation of the gradient with only a noisy numerical observation of the utilities.

Most of previous works on DFSO consider a control center that updates the entire action vector during the algorithm, see [30] for more details. However, in our distributed setting, each node is only able to update its own action. Nevertheless, a stochastic approximation method using the simultaneous perturbation gradient approximation (SPGA) [31] can be an option to solve our distributed DFSO problem. The SPGA algorithm was initially proposed to accelerate the convergence speed of centralized multi-variate optimization problems with deterministic objective functions. Two measurements of the objective function are needed per update of the action. The approximation of the partial derivative of an element $i$ is given by

$$\hat{g}_{i,k} = \frac{f(a_k + \gamma_k \Delta_k) - f(a_k - \gamma_k \Delta_k)}{2\gamma_k \Delta_{i,k}},$$  
(2)

where $\gamma_k > 0$ is vanishing and $\Delta_k = [\Delta_{1,k}, \ldots, \Delta_{N,k}]$. Each element $\Delta_{i,k}$ is an i.i.d. variable with zero mean. Two successive measurements of the objective function are required to perform a single estimation of the gradient. The interest of the SPGA method is that each variable can be updated simultaneously and independently. Spall has also proposed a one-measurement version of the SPGA algorithm in [32] with

$$\hat{g}_{i,k} = \frac{f(a_k + \gamma_k \Delta_k)}{2\gamma_k \Delta_{i,k}}.$$  
(3)

This algorithm also leads $a_k$ to converge, but with a decreased speed compared with the two-measurement SPGA. An essential result is that the estimation of gradient using (2) or (3) is unbiased if $\gamma_k$ is vanishing, as long as the objective function $f$ is static. However, if the objective function is stochastic and its observation is noisy, there would be an additional term of stochastic noise in the numerator of (2) and (3). This may seriously affect the performance of approximation when the value of $\gamma_k$ is too small. As a consequence, the SPGA algorithm cannot be used directly to solve our stochastic optimization problem.

The authors in [33] proposed a fully distributed Nash equilibrium seeking algorithm which requires only a measurement of the numerical value of the static utility function. Their scheme is based on deterministic sine perturbation of the payoff function in continuous time. In [34], the authors extended the work in [33] to the case of discrete time and stochastic state-dependent utility functions. The convergence of the proposed method to within a close region around Nash equilibrium has been proved. However, in a distributed setting, it is challenging to ensure that the sine perturbation of different nodes satisfies the orthogonality requirement, especially when the number of nodes is large. Moreover, the continuous sine perturbation based algorithm converges slowly in a discrete-time system. A stochastic perturbation based algorithm has also been proposed in [35] to solve optimization problems. The algorithm is given by

$$a_{k+1} = a_k + \beta v_k f(a_k + v_k),$$  
(4)

with $v_k$ is a zero-mean stochastic perturbation. The behavior of (4) has been analyzed in [35], under the assumption that the objective function is static and quadratic. Our proposed algorithm is different from (4) as we use a random perturbation with vanishing amplitude. In addition, the objective function is stochastic with non-specified form in our setting, which is much more challenging. Furthermore, we consider a situation where nodes have to exchange their local utilities to estimate the global utility and each node may have information of part of the local utilities of other nodes.

As we have mentioned in Section I, DFSO has attracted much interests in learning community. Both [15] and [13] used a single realization of the objective function to estimate the gradient, in a way similar to the gradient estimator in (4). The major difference is that the cost function $f(a_k; s_k)$ is affected by some stochastic parameter or oblivious adversary $s_k$, with $s_k$ independent of $a_k$. The algorithm in [15] has been shown to be near-optimal after a total number of $K$ iterations. More precisely, it was shown that the resulting expected regret $\mathbb{E} \left[ \sum_{k=1}^{K} f(a_k; s_k) \right]$ is $O(K^{3/4})$, under the assumption that $f$ is convex and Lipschitz. However, finite-time horizon is not adapted to wireless networks, as it is usually hard to predict the duration of connections and the total number of iterations. For the above reason, in this work, we aim to propose an optimal solution with asymptotic performance optimality. Moreover, a control center is required in the algorithm proposed in [15] in order to generate a random unit norm perturbation vector at each iteration. However, a unit norm perturbation vector is hard to perform in a distributed network since in this case each node must generate independently its own random perturbation.

In [13], a quadratic objective function $f$ is specifically considered. It was shown that $\hat{g}_k$ is an unbiased estimator.
of gradient in this special case. Another interesting work is the one in [16] that also used a single-point gradient estimator, and applied the technique of self-concordant barrier to accelerate the convergence. It was shown that he achievable expected regret is $O\left(K^{1/2}\right)$ when $f$ is smooth and strongly-convex. Again, finite-time horizon was considered in [16] and a controller center is needed to generate random unit norm vectors and to handle some matrix operations.

There are also numbers of papers based on two-point gradient estimator, e.g., [14], [36], [37]. These works assume that it is possible to observe two values of the objective function $f(a_k^1; s_k)$ and $f(a_k^2; s_k)$ under the same realization $s_k$ of the stochastic environment state. However this assumption may be unrealistic in our setting where the value of $s_k$ changes fast. This implies that it is practically impossible to observe two network utilities using different action $a_k$ under the same environment state. For this reason, we focus on single-point gradient estimator in this paper.

III. System Model

This section presents the problem formulation as well as the basic assumptions. Throughout this paper, matrices and vectors are in boldface letters. Calligraphic font denotes set. $\|\mathbf{x}\|$ denotes the Euclidean norm of any vector $\mathbf{x}$.

A. Problem Formulation

Consider a network consisting of a finite set of nodes $\mathcal{N} = \{1, \ldots, \mathcal{N}\}$. Each node $i$ controls its own action $a_{i,k}$ at each discrete timeslot $k$, in order to maximize the performance of the network. $a_{i,k}$ is assumed to be a continuous variable. Introduce the action vector $\mathbf{a}_k = [a_{1,k}, \ldots, a_{\mathcal{N},k}]^T$ which contains the actions of all nodes at timeslot $k$. Let $\mathcal{A}_i$ denote the feasible action set of node $i$, i.e., $a_{i,k} \in \mathcal{A}_i$. Introduce $\mathcal{A} = \mathcal{A}_1 \times \ldots \times \mathcal{A}_\mathcal{N}$.

In general, the local utility of node $i$ at timeslot $k$ depends on the action of a subset of nodes $\mathcal{N}_{i,k} \subseteq \mathcal{N}$, as well as some stochastic environment state (e.g., random channels between nodes in wireless networks) that can be described by a random vector $\mathbf{s}_k \in \mathcal{S}_i$. Note that for any $i \neq j$, $\mathcal{S}_{i,k}$ and $\mathcal{S}_{j,k}$ may share some common elements. The local utility function of node $i$ can then be denoted as $u_i([a_{j,k}]_{j \in \mathcal{N}_{i,k}}; \mathbf{s}_k)$. We are interested in the global utility $f(\mathbf{a}_k; \mathbf{s}_{1,k}, \ldots, \mathbf{s}_{\mathcal{N},k}) : \mathcal{A} \times \mathcal{S}_1 \times \ldots \times \mathcal{S}_\mathcal{N} \to \mathbb{R}$ of the network, which is defined as the sum of the local utilities of all nodes, i.e.,

$$f(\mathbf{a}_k; \mathbf{s}_{1,k}, \ldots, \mathbf{s}_{\mathcal{N},k}) = \sum_{i \in \mathcal{N}} u_i([a_{j,k}]_{j \in \mathcal{N}_{i,k}}; \mathbf{s}_k). \tag{5}$$

It is worth mentioning that we are considering a black-box optimization problem in this paper. The closed form expression of $u_i$ is unavailable, and the exact knowledge of $\mathcal{N}_{i,k}$ and $\mathbf{s}_k$ may also be unknown. Instead of using the heavy notations as in (5), we will denote throughout this paper the local utility and global utility as $u_i(a_k; \mathbf{s}_k)$ and $f(\mathbf{a}_k; \mathbf{s}_k)$ respectively, where $\mathbf{s}_k = [\mathbf{s}_{j,k}]_{j \in \mathcal{N}_k}$ is a collection of all elements of $\mathbf{s}_k$ for any $j \in \mathcal{N}$. In fact $u_i(a_k; \mathbf{s}_k)$ is a general notation, it implies that $u_i$ is potentially affected by $a_k$ and $\mathbf{s}_k$, but does not necessarily mean that $u_i$ is affected by every element of $a_k$ and $\mathbf{s}_k$. More importantly, these notations have no influence on our theoretical results.

With the simplified notations, we can rewrite (5) as

$$f(a_k; \mathbf{s}_k) = \sum_{i \in \mathcal{N}} u_i(a_k; \mathbf{s}_k).$$

Here the random term $\mathbf{s}_k$ is considered as an i.i.d. ergodic stochastic process. The network performance is characterized by the average global utility $F(\mathbf{a}) : \mathcal{A} \to \mathbb{R}$ with

$$F(\mathbf{a}) = \mathbb{E}_S [f(\mathbf{a}; \mathbf{S})].$$

In this work, we consider a challenging setting where nodes do not have the knowledge of $\mathbf{s}_k$ nor the closed-form expression of the utility functions. Each node $i$ has only a numerical estimation $\tilde{u}_{i,k} \in \mathbb{R}$ of its local utility $u_i(a_k; \mathbf{s}_k)$ at each timeslot. Assume that

$$\tilde{u}_{i,k} = u_i(a_k; \mathbf{s}_k) + \eta_{i,k}, \tag{6}$$

with $\eta_{i,k} \in \mathbb{R}$ some additive noise. Each node can receive the local utilities of some (not all) other nodes. Furthermore, two different nodes may receive utilities from two different subset of nodes. This will be discussed in Section III.C where we provide the exact communication model between nodes.

In summary, each node has an incomplete information of the global utility value at each time. In this challenging context, the objective is to develop a distributed solution to the following problem

$$\begin{align*}
\text{maximize} & \quad F(\mathbf{a}) = \mathbb{E}_S [f(\mathbf{a}; \mathbf{S})] \\
\text{subject to} & \quad a_i \in \mathcal{A}_i, \ \forall i \in \mathcal{N}.
\end{align*} \tag{7}$$

The solution must be obtained by using the available information of the global utility value at each time, and the gradient of $f$ is neither available nor possible to derive as explained earlier in this paper. An application example is introduced in Section VII-A to highlight the interest of this problem.

The objective function $F(\mathbf{a})$ is assumed to be a strictly concave function of $\mathbf{a} \in \mathcal{A}$. Note that in the analysis of the convergence Rate in Section VI, an additional assumption concerning the strong concavity of the objective function will be considered. We denote $\mathbf{a}^\star$ as the solution of problem (7) throughout this paper. The formal assumptions on the utility functions, objective function $F$ and optimal solution $\mathbf{a}^\star$ will be provided in Section III.B. In the rest of the paper, we use the following notations to represent the partial derivatives and the gradient of the objective functions:

$$g_i(\mathbf{a}; \mathbf{S}) = \frac{\partial f}{\partial a_i}(\mathbf{a}; \mathbf{S}),$$

$$G_i(\mathbf{a}) = \mathbb{E}_S [g_i(\mathbf{a}; \mathbf{S})] = \frac{\partial F}{\partial a_i}(\mathbf{a}),$$

$$G(\mathbf{a}) = [G_1(\mathbf{a}), \ldots, G_\mathcal{N}(\mathbf{a})]^T = \nabla F(\mathbf{a}).$$

B. Basic Assumptions

We present the basic assumptions on the objective and utility functions considered in this paper in order to guarantee the performance of our proposed algorithm.
First of all, we have the following assumption on the concavity of \( F \).

**Assumption 1:** Both first-order and second-order partial derivatives of \( F(a) \) exist continuously. The objective function \( F(a) \) is strictly concave, i.e.,

\[
(a - a')^T \cdot (G(a) - G(a')) < 0, \quad \forall a, a' \in A : a \neq a' \quad (8)
\]

There exists a unique \( a^* \in A \) such that \( a^* \) is not on the boundary of \( A \) and \( G(a^*) = 0 \).

It is worth mentioning that in the convergence rate analysis, an additional assumption of strong concavity will be used. This will be presented in Section VI as it is not required neither in the development of the algorithm nor in its almost sure convergence proof.

**Assumption 2:** The gradient \( G(a) \) is Lipschitz, i.e., there exists a constant \( L_G \in \mathbb{R}^+ \) such that

\[
\|G(a) - G(a')\| = \|\nabla F(a) - \nabla F(a')\| \\
\leq L_G \|a - a'\|, \quad \forall a, a' \in A, \quad (9)
\]

In other words, \( F(a) \) is \( L_G \)-smooth.

Note that Assumption 2 is less restrictive than the assumption that the second-order derivative of \( F \) is bounded as considered in the previous work [1]. Because \( \|G(a) - G(a')\| \leq N \alpha_G \|a - a'\| \) can be deduced from \( \frac{\partial^2}{\partial a_i \partial a_j} F(a) \leq \alpha_G < +\infty, \forall i, j \), while the converse is not true.

We have some further assumptions on the local utility functions, which will be useful in our analysis.

**Assumption 3:** For any \( i \in \mathcal{N} \), there exists a bounded constant \( \alpha_i \in (0, \infty) \) such that \( \sqrt{\mathbb{E}_s} [u_i^2(a; S)] \leq \alpha_i, \quad \forall a \in A \).

Finally, we consider a common assumption on the noise term \( \eta_{i,k} \) introduced in (6).

**Assumption 4:** The noise \( \eta_{i,k} \) is zero-mean, uncorrelated, and has a bounded variance: i.e., for any \( i \in \mathcal{N} \), we have \( \mathbb{E} [\eta_{i,k}] = 0, \mathbb{E} [\eta_{i,k}^2] = \sigma_{\eta}^2 < \infty \), and \( \mathbb{E} [\eta_{i,k} \eta_{j,k}] = 0 \) for any \( i \neq j \).

Note that Assumptions 1-4 are standard. There exist many functions satisfying these assumptions, e.g.,

- \( u_{i}^{eg1}(a; S) = \omega_1 \log \left( \frac{a_i \sum s_{j,k} e^{a_j}}{s_{i,j} e^{a_i}} \right) - \omega_2 e^{a_i} \), where \( -\infty < a_{\min} \leq a_i \leq a_{\max} < \infty \) for any \( i \in \mathcal{N} \) and each \( s_{i,j} \) follows independent exponential distribution. \( \omega_1, \omega_2 \) and \( \sigma^2 \) are positive constants;
- \( u_{i}^{eg2}(a; S) = -s_i a_i^2 + 4 \log(a_i) + \frac{1}{2(N-1)} \sum_{j \neq i} a_i a_j \), where \( 0 < a_{\min} \leq a_i \leq a_{\max} < \infty \) for any \( i \in \mathcal{N} \) and each \( s_i \in [0.5, 1.5] \) follows independent uniform distribution.

It is straightforward to show that \( \mathbb{E}_s [u_{i}^{eg1}]^2 \) and \( \mathbb{E}_s [u_{i}^{eg2}]^2 \) are bounded. We can also prove easily that the corresponding average global utilities of \( u_{i}^{eg1} \) and \( u_{i}^{eg2} \) are strictly concave and smooth. In fact, both examples are considered to obtain simulation results in this paper. Please refer to Section VII for more details.

C. Network Model and Exchange of Local Utilities Among Nodes

In this paper, we consider a realistic network model where a node has only the knowledge of the local utilities of a subset \( I_{i,k} \) of the other nodes, with \( I_{i,k} \subseteq \mathcal{N} \setminus \{i\} \). Notice that we do not assume any specified network topology and each node has a different and independent set \( I_{i,k} \). This implies that two different nodes may receive at a given time the local utilities from two different subsets of nodes. We present a toy example first to better explain our model and highlight further the difficulty of the optimization problem.

**Example 1:** Here we consider a D2D network with three links. As described in Figure 1, at time \( k \), each transmitter receives the feedback (i.e. numerical value of the local utility) from its own receiver. Note that the received local utility by each transmitter can be a noisy estimation of the real local utility as given by Equation (6). Transmitter 1 knows then \( \tilde{u}_{1,k} \), transmitter 2 knows \( \tilde{u}_{2,k} \), and transmitter 3 knows \( \tilde{u}_{3,k} \). Then, transmitter 1 receives the local utility of link 3 from transmitter 3, transmitter 2 receives the local utility of link 1 transmitter 1, and transmitter 3 receives the one of link 2 from transmitter 2. The ideal estimated global utility should be \( \hat{f}_k = \tilde{u}_{1,k} + \tilde{u}_{2,k} + \tilde{u}_{3,k} \), but each transmitter only knows the local utilities of part of the links of the network. Node 1 (transmitter 1) has to evaluate an incomplete information of the global utility \( f_k \) since this node knows only \( \tilde{u}_{1,k} \) and \( \tilde{u}_{3,k} \). Similarly, node/transmitter 2 knows only \( \tilde{u}_{2,k} \) and \( \tilde{u}_{1,k} \), and node 3 knows only \( \tilde{u}_{2,k} \) and \( \tilde{u}_{3,k} \). One can see that the nodes/transmitters have different knowledge of local utilities and each node must decide independently from other nodes of its own action \( a_{i,k} \). Please note that, in some situations, the transmitters do not have to exchange local utilities: this may be the case when each receiver can send its feedback to part of the transmitters (including its own). For example, receiver 1 sends \( \tilde{u}_{1,k} \) to transmitters 1 and 2, receiver 2 sends \( \tilde{u}_{2,k} \) to transmitters 2 and 3, and receiver 3 sends \( \tilde{u}_{3,k} \) to transmitters 3 and 1.

Throughout this paper, we have the following assumption that defines the subset \( I_{i,k} \) for each node \( i \) at each time/iteration \( k \).

**Assumption 5:** At any iteration \( k \), an arbitrary node \( i \) knows the utility \( \tilde{u}_{i,k} \) of another node \( j \) with a constant probability \( p \in (0, 1] \). This means that the elements contained in the set \( I_{i,k} \) are random, and for any \( j \neq i \) we have

\[
\mathbb{P}[j \in I_{i,k}] = p, \quad \mathbb{P}[j \notin I_{i,k}] = 1 - p. \quad (10)
\]
IV. DISTRIBUTED OPTIMIZATION ALGORITHM USING STOCHASTIC PERTURBATION

We describe our optimization algorithm in this section. We start with the approximation of the value of the global utility (objective function) in Section IV-A. Then, we present our DOSP algorithm (namely distributed optimization algorithm using stochastic perturbation) in Section IV-B. An application example of DOSP is also provided in Section IV-C.

A. Estimation of Global Utility

Recall that each node $i$ knows only the local utilities of a subset $\mathcal{I}_{i,k} \subseteq \mathcal{N} \setminus \{i\}$ of the other nodes at each time $k$. Based on the incomplete knowledge of local utilities, each node can estimate the global utility as follows

$$\tilde{f}_{i,k} = \begin{cases} \tilde{u}_{i,k} + \frac{N-1}{|\mathcal{I}_{i,k}|} \sum_{j \in \mathcal{I}_{i,k}} \tilde{u}_{j,k}, & \text{if } |\mathcal{I}_{i,k}| \neq 0, \\ 0, & \text{if } |\mathcal{I}_{i,k}| = 0. \end{cases}$$ (11)

The basic idea is to consider $(N-1) \sum_{j \in \mathcal{I}_{j,k}} \tilde{u}_{j,k}/|\mathcal{I}_{i,k}|$ as a surrogate function of $\sum_{j \notin N \setminus \{i\}} \tilde{u}_{j,k}$ when $\mathcal{I}_{i,k}$ is non-empty. Otherwise, $\tilde{f}_{i,k}$ is set as 0 since node $i$ does not know any utility of the other nodes and cannot estimate the global utility of the system. Note that different nodes may have different knowledge of the global utility as $\mathcal{I}_{i,k}$ is independent for each node $i$. For example, node $i$ may know $\tilde{u}_{j,k}$ of a different node $j$, whereas node $j$ may not know $\tilde{u}_{i,k}$.

Note that $\tilde{f}_{i,k}$ defined by (11) is affect by the random terms $\Phi_{i,k}$, $\eta_{i}$, and $\mathcal{I}_{i,k}$. The following lemma states that $\tilde{f}_{i,k}$ is a reasonable estimator of the global utility.

**Lemma 1**: Under Assumptions 4 and 5, the expected value of $\tilde{f}_{i,k}$ over $\mathcal{I}_{i,k}$ and $\eta_{i}$ is proportional to the actual global utility $f(\alpha; S_{k})$, i.e.,

$$\mathbb{E}_{\mathcal{I}_{i,k}} \left[ \tilde{f}_{i,k}(\alpha; S_{k}, \eta_{i}, \mathcal{I}_{i,k}) \right] = q f(\alpha; S_{k})$$ (12)

where the constant $q$ is given by

$$q = \mathbb{P}[|\mathcal{I}_{i,k}| \neq 0] = 1 - (1 - p)^{N-1}.$$ (13)

**Proof**: See Appendix A.

B. Algorithm

The DOSP algorithm is presented in Algorithm 1. The algorithm is distributed since each node updates its action independently and each node may not know the local utilities of all the other nodes.

**Algorithm 1** DOSP Algorithm For Each Node $i$

1) Initialize $k = 1$ and set the action $a_{i,1}$ randomly.
2) Generate a random variable $\Phi_{i,k}$, and perform action $a_{i,k} + \gamma_{k}\Phi_{i,k}$.
3) Estimate $\tilde{u}_{i,k}$, exchange its value with some other nodes, and calculate $\tilde{f}_{i,k}$ using (11) based on the collected local utilities.
4) Update $a_{i,k+1}$ according to equation (14)-(15).
5) $k = k + 1$, and go to 2.

At each iteration $k$, an arbitrary reference node $i$ updates its action by applying

$$\bar{a}_{i,k+1} = a_{i,k} + \beta_{k} \Phi_{i,k} \tilde{f}_{i,k},$$ (14)

$$a_{i,k+1} = \min \left\{ \max \left\{ \bar{a}_{i,k+1}, a_{i,min} + \alpha_{\Phi} \gamma_{k+1} \right\}, a_{i,max} - \alpha_{\Phi} \gamma_{k+1} \right\},$$ (15)

in which $\bar{a}_{i,k+1}$ represents an intermediate variable, $\beta_{k}$ and $\gamma_{k}$ are vanishing step-sizes, $\Phi_{i,k}$ is a zero-mean random variable generated independently by each node $i$, and $\Phi_{k} = [\Phi_{1,k}, \ldots, \Phi_{N,k}]^{T}$. We present some desirable properties of $\beta_{k}$, $\gamma_{k}$, and $\Phi_{i,k}$ in the following assumption.

**Assumption 6**: Both $\{\beta_{k}\}$ and $\{\gamma_{k}\}$ are real positive vanishing sequences that satisfy

$$\sum_{k=1}^{\infty} \beta_{k} \gamma_{k} = \infty \text{ and } \sum_{k=1}^{\infty} \beta_{k}^{2} < \infty.$$ (16)

The elements of $\Phi_{k}$ follow some i.i.d. symmetric probability distribution, so that $\mathbb{P}[\Phi_{i,k} \geq c] = \mathbb{P}[\Phi_{i,k} \leq -c]$ for any $c \in \mathbb{R}^{+}$. We have $\mathbb{E}[\bar{a}_{i,k}] = \mathbb{E}[\Phi_{i,k}^{2}] = 0$, $\mathbb{E}[\Phi_{i,k}^{2}] = \sigma_{\Phi}^{2}$, and $\mathbb{E}[\Phi_{i,k} \Phi_{j,k}] = 0$, $\forall i \neq j$. There exist $\alpha_{\Phi} > 0$ such that $\mathbb{P}[\Phi_{i,k} \leq \alpha_{\Phi}, \forall i \in \mathcal{N}]$.

Recall that the approximation $\tilde{f}_{i,k}$ of the global utility is calculated by each node using (11), in which the value depends on the actual action performed by each node $\bar{a}_{i,k} = a_{i,k} + \gamma_{k}\Phi_{i,k}$ and the stochastic term $S_{k}$. We can see that $\mathbb{E}_{\Phi}[\bar{a}_{i,k}] = \mathbb{E}_{\Phi}[a_{i,k}]$ and the difference between $\bar{a}_{i,k}$ and $a_{i,k}$ decreases with $\gamma_{k}$. At each iteration, we also need to ensure that the action $\bar{a}_{i,k}$ is feasible, i.e., $\bar{a}_{i,k} \in [a_{i,min}, a_{i,max}]$. Hence, we need to project step (15) which leads to $a_{i,k+1} + \gamma_{k+1} [\Phi_{i,k+1}] \in [a_{i,min}, a_{i,max}]$ for any $i \in \mathcal{N}$, as $|\Phi_{i,k}| \leq \alpha_{\Phi}$ by Assumption 6.

An example is presented in Section IV-C to describe in detail the application of Algorithm 1 in practice.

Clearly, (14) can be written in the general form (1), with

$$g_{i,k} = \Phi_{i,k} \tilde{f}_{i,k}.$$ (17)

By introducing $g_{k} = [\hat{g}_{1,k}, \ldots, \hat{g}_{N,k}]^{T}$, we have $a_{k+1} = a_{k} + \beta_{k} \hat{g}_{k}$. Here $\hat{g}_{k}$ can be seen as an estimation of the gradient $G(a_{k}) = \nabla F(a_{k})$. It is worth mentioning that such an estimation is biased in general. Nevertheless, $\hat{g}_{k}$ could be an asymptotically unbiased estimation of $G(a_{k})$ when $k \to \infty$, as long as the parameters $\gamma_{k}$ and $\Phi_{k}$ satisfy Assumption 6. The detailed discussion will be provided in Section V-A.

The conditions on the parameters $\gamma_{k}$ and $\Phi_{k}$ can be easily achieved. We show in Example 2 a common setting, which is also used to obtain the simulation results presented in Section VII.

**Example 2**: An easiest choice of the probability distribution of $\Phi_{i,k}$ is the symmetrical Bernoulli distribution with $\Phi_{i,k} \in \{-1,1\}$ and $\mathbb{P}[\Phi_{i,k} = 1] = \mathbb{P}[\Phi_{i,k} = -1] = 0.5$, $\forall i \in \mathcal{N}$. We can verify the conditions in Assumption 6 with $\sigma_{\Phi}^{2} = \alpha_{\Phi} = 1$.

Let $\beta_{k} = \beta_{0}^{k-\nu_{1}}$ and $\gamma_{k} = \gamma_{0}^{k-\nu_{2}}$ with the constants $\beta_{0}, \gamma_{0} \in \mathbb{R}^{+}$, so that both $\beta_{k}$ and $\gamma_{k}$ are vanishing. We can see that $\sum_{k=1}^{\infty} \beta_{k}^{2}$ converges if $\nu_{1} > 0.5$, and $\sum_{k=1}^{\infty} \beta_{k} \gamma_{k}$ converges if $\nu_{1} + \nu_{2} < 1$. Clearly, (16) is satisfied if $0.5 < \nu_{1} \leq 1$ and $0 < \nu_{2} < 1 - \nu_{1} < 0.5$. Thus we show that (14) and (15) are satisfied.
with transmission power $\Phi_{i,k}$, utility $u_{i,k}$, and some probability. The red links represent the transmission failure. Note that in some situations transmitters can also exchange their received $\tilde{u}_{i,k}$, please refer to Figure 1 as an example.

Remark 1: The proposed algorithm has a similar shape compared with other existing methods proposed in [13], [15], [16], [33]–[35]. The difference between our solution and the sine perturbation based method [33], [34] is that we use a random vector $\Phi_k$ instead of some deterministic sine functions as the perturbation term. Furthermore, the amplitude of random perturbation is vanishing in our algorithm, which is not the case in the algorithm presented in [13], [15], [35].

C. Application Example

This section presents the application of Algorithm 1 to perform power control in a D2D network, in order to highlight the interest of our solution. We focus on the requirement arisen by Algorithm 1, in terms of computation, memory and informational exchange.

Figure 2 briefly shows the algorithm procedure during one iteration. Recall that $\hat{a}_{i,k}$ denotes the actual value of the action set by transmitter $i$ at iteration $k$. In order to update $\hat{a}_{i,k}$, each transmitter $i$ needs to

- independently and randomly generate a scalar $\Phi_{i,k}$ that satisfies Assumption 6;
- use a pre-defined vanishing sequence $\gamma_k$ which is common for each link;
- independently update $a_{i,k}$ by applying (14) and (15); more details on this step will be provided later in this section.

Then $\hat{a}_{i,k}$ is given by $\hat{a}_{i,k} = a_{i,k} + \gamma_k \Phi_{i,k}$.

Each transmitter $i$ transmits to its associated receiver with the transmission power of value $\hat{a}_{i,k}$. The associated receiver $i$ estimate the numerical value of its local utility $\tilde{u}_{i,k}$ and send this value to transmitter $i$ as a feedback. $\tilde{u}_{i,k}$ is a noisy version of the exact local utility of the link due to the wireless channel and/or imperfect feedback. Each transmitter can evaluate its $\tilde{f}_{i,k}$ using (11) under Assumption 5 that every transmitter only receives feedback from part of the receivers.

Then iteration $k + 1$ starts, and each transmitter $i$ needs to update its power allocation strategy. For that, the transmitter $i$ will perform the following steps:

- use a pre-defined vanishing sequence $\beta_k$;
- reuse the local random value $\Phi_{i,k}$, which was generated by transmitter $i$ at iteration $k$. It means that the value of $\Phi_{i,k}$ should be saved temporarily;
- use the numerical value of $\tilde{f}_{i,k}$, which has been explained already.

Each transmitter can then update $a_{i,k+1}$ independently using (14) and (15).

Then, each transmitter updates $\hat{a}_{i,k+1}$ in the same way as the previous iteration, i.e., $\hat{a}_{i,k+1} = a_{i,k+1} + \gamma_{k+1} \Phi_{i,k+1}$, with $\Phi_{i,k+1}$ a newly generated pseudo-random value.

We can see that Algorithm 1 can be easily applied in a network of multiple links: the algorithm itself has a low complexity and each receiver only needs to feedback one quantity ($\tilde{u}_{i,k}$) per iteration to perform the algorithm.

V. Almost Sure Convergence

This section investigates the asymptotic behavior of Algorithm 1. We can show that the divergence

$$d_k = \|a_k - a^*\|^2$$

which measures the difference between the actual action $a_k$ and the optimal action $a^*$, tends to zero almost surely (a.s.) when $k \to \infty$. Before proving this result, we need first to show that our DOSP algorithm provides a biased estimation of the gradient. This will be useful in the almost sure convergence proof.

A. Biased Estimation of Gradient

This section presents our first result in Theorem 1 to explain a main property of our DOSP algorithm: how $\tilde{g}_k$, given by (17), can be a reasonable estimator of $G(a_k)$ = $\nabla F(a_k)$.

**Theorem 1:** Suppose that each node $i$ updates its action $a_{i,k}$ using (14). Define

$$\tilde{g}_k = E_S \mathbf{x} \mathbf{u} I [\tilde{g}_k]$$

$$b_k = (\sigma^2_{\tilde{g}}(\gamma_k))^{-1} \tilde{g}_k - G(a_k)$$

then under Assumptions 1, 2, 4, 5, and 6, we have $b_k \neq 0$ in general, and

$$\|b_k\| \leq \frac{\alpha_k}{2} L G N^2 \gamma_k$$

implying that $\|b_k\| \to 0$ as $k \to \infty$ if $\gamma_k$ is vanishing.

**Proof:** See Appendix B.

Note that an alternative way to write (19) is $\tilde{g}_k = E [\tilde{g}_k | a_k]$ since the expectation is taken over all stochastic terms expect $a_k$. In this paper, we prefer to highlight the stochastic terms in (19) for clarity. The estimation bias of the gradient is characterized by $b_k$. From Theorem 1, we conclude that, for a general concave and smooth objective function $F$, $\|b_k\| \to 0$ can be guaranteed by using a vanishing sequence $\gamma_k$. Hence $\tilde{g}_k$ defined in Algorithm 1 is a reasonable estimator of $G(a_k)$.

Note that Theorem 1 presents a more general result compared with Lemma 1 in our previous work [1], in which we had $\|b_k\| \leq 2^{-1} \alpha' \alpha \sigma N^2 \gamma_k$ under a more restrictive assumption that $\frac{\sigma^2_{\tilde{g}}(\gamma_k)}{\alpha_k} \leq \alpha'$, $\forall i, j \in N$. 

In general $b_k \neq 0$, while there does exist one special class of $F$ resulting to $b_k = 0, \forall k$. This case arises when $F$ is a quadratic function.

**Corollary 1:** Suppose that $F$ is a quadratic function and Assumptions 4-6 hold, then $b_k = 0, \forall k$.

**Proof:** See Appendix C.

The proof of Corollary 1 is also given in Appendix B. This result implies that our DFSO algorithm provides an unbiased gradient estimation in this special case. To the best of our knowledge, a quadratic objective function $F$ is the only case where an unbiased gradient estimator can be obtained by DFSO. In fact, $b_k = 0$ holds in this case even with a constant step-size $\gamma_k = \gamma_0$. Such a special case has been well studied in [13]. In this work, we mainly focus on the general strictly concave and smooth function.

**B. Almost Sure Convergence**

In order to proceed with the proof of convergence, we first rewrite (14) in the generalized Robbins-Monro form [25], i.e.,

$$
\tilde{a}_{k+1} = a_k + \beta_k \tilde{g}_k = a_k + \beta_k b_k \sigma_a^2 q_k^2 \nabla F(a_k) + \beta_k (\tilde{g}_k - \bar{g}_k) = a_k + \sigma_a^2 q_k^2 \tilde{e}_k + \beta_k e_k,
$$

where $\bar{g}_k$ and $b_k$ are defined in (19) and (20) respectively. $e_k$ can be seen as the zero-mean stochastic noise, which is the difference between the value of a single realization of $\tilde{g}_k$ and its average $\bar{g}_k$ as defined in (19), i.e.,

$$
e_k = \tilde{g}_k - \bar{g}_k.
$$

**Remark 2:** The analysis presented in this work is challenging and different from the existing results. An explicit difference comes from the unique feature of the algorithm itself as discussed in Remark 1: we are using a different method to estimate the gradient. Besides, the objective function is stochastic with general form in our problem, while it is considered as static in [31], [32] and it is assumed to be static and quadratic in [35]. The existing work in learning community focused on the non-asymptotic convergence rate, while the almost surely convergence are not considered.

The general property of $b_k$ has been discussed in Theorem 1. To perform the analysis of convergence, we still need the following lemmas concerning the property of $\tilde{g}_k$ and $e_k$.

**Lemma 2:** Under Assumptions 3-6, we have $E_s, \psi, \eta, \pi \| \bar{g}_k \|^2 \leq \alpha_g^2$ with

$$
\alpha_g^2 = \left( (\psi + \gamma) (N - 1)^2 + p (N - 1) + q \right) \frac{\alpha_a^2 \sigma_a^2}{N} + \left( (N - 1)^2 \psi + q \right) \frac{\sigma_a^2 \sigma_q^2}{N^2}.
$$

in which $q$ is defined by (13) and

$$
\psi = \sum_{n=1}^{N-1} \frac{1}{n} p^n (1 - p)^{N - 1 - n} \left( \frac{N - 1}{n} \right).
$$

**Proof:** See Appendix D.

Note that the closed form expression of $\psi$ is hard to derive. Note that it is easy to get the following upper bound of $\psi$:

$$
\psi < \sum_{n=1}^{N-1} P [ |I_n| = n ] = q \frac{1}{n} \leq 1 \text{ with } 1 \leq n \leq N - 1.
$$

However, in order to keep the upper bound $E_s, \psi, \eta, \pi \| \bar{g}_k \|^2 \leq \alpha_g^2$ as tight as possible, we decided to use the expression in (24).

Based on Lemma 2, we can prove the convergence of $\sum \beta_k e_k$ by applying Doob’s martingale inequality.

**Lemma 3:** If Assumptions 3-6 are satisfied, then for any constant $\rho > 0$, we have

$$
\lim_{K \to \infty} P \left( \sup_{K' \geq K} \left\{ \sum_{k=K}^{K'} \beta_k e_k \right\} \geq \rho \right) = 0, \quad \forall \rho > 0.
$$

**Proof:** See Appendix E.

According to the results in Theorem 1, Lemma 2, and Lemma 3, we can build conditions under which $a_k \to a^*$ a.s..

**Theorem 2:** If Assumptions 1-6 are satisfied, then $a_k \to a^*$ almost surely as $k \to \infty$ by applying Algorithm 1.

**Proof:** See Appendix F.

**VI. CONVERGENCE RATE**

In this section, we study the average convergence rate of the proposed algorithm in order to investigate how fast the proposed algorithm converge to the optimum from a quantitative point of view. The analysis also provides a detailed guideline of setting the parameters $\beta_k$ and $\gamma_k$ which determine the convergence rate.

Before starting the analysis, we stress out that all the previous assumptions on the objective function and step-sizes $\beta_k$ and $\gamma_k$ (i.e. Assumptions 1-6) hold true in this section. In addition, we consider the following two assumptions:

- We assume that $\beta_k / \gamma_k \to 0$ when $k \to 0$
- We assume that the objective function is strongly concave. More precisely, this assumption is given as follows.

**Assumption 7:** $F(a) \to \kappa-$strongly concave, i.e., there exists $\kappa > 0$ such that

$$
(a - a^*)^T \cdot \nabla F(a_k) \leq -\kappa \|a - a^*\|^2, \quad \forall a \in A.
$$

We start with the analysis considering general forms of vanishing step-sizes $\beta_k$ and $\gamma_k$ in Section VI.A. We then consider special forms of these parameters in Section VI.B, which leads us to obtain the optimal convergence rate order of $O(K^{-0.5})$. A toy example is then considered afterwards to further illustrate our results.

**A. Analysis for General Forms of Vanishing $\beta_k$ and $\gamma_k$**

To study the converge rate, we have first to consider the evolution of the average divergence $D_k = E[\|a_k - a^*\|^2]$. An essential step of the analysis is to investigate the relation between the divergence values obtained at two successive iterations. Our result is stated in Lemma 4. Recall that $\alpha_g^2$ defined in (24) is the upper bounds of $E[\|\bar{g}_k\|^2]$.

**Lemma 4:** Introduce an integer $K_c$ as the minimum value of $k$ such that $\alpha_g^2 \in [a_{\text{min}} + \alpha\gamma \gamma_k, a_{\text{max}} - \alpha\gamma \gamma_k], \forall \gamma_k \in N$. Suppose that Assumptions 1-7 hold, then for any $k \geq K_c$:

- $D_k$ resulting from Algorithm 1 is such that

$$
D_{k+1} \leq (1 - A\beta_k \gamma_k) D_k + B\beta_k \gamma_k^2 \sqrt{D_k} + C\beta_k^2,
$$
with the constants
\[ A = 2q^2A \kappa, \quad B = q_0A^2L_2 N^2, \quad C = \alpha^2. \]  
(29)

- If we consider an additional assumption that the objective function \( F \) is quadratic, then we have
\[ D_{k+1} \leq (1 - 2q^2A \kappa \beta \gamma_k) D_k + \alpha^2 \beta^2_k. \]  
(30)

**Proof:** See Appendix G.

**Remark 3:** In the special case where \( F \) is quadratic, \( b_k = 0 \) by Corollary 1. It is easy to deduce that \( D_k = O(k^{-1}) \) from (30). Thus the optimal convergence rate \( D_k = O(k^{-1}) \) can be achieved by using \( \beta_k = 0 \) and \( \gamma_k = 0 \). In the rest of this section, we focus on the general setting of \( F \).

It is worth mentioning that we have a nonzero term as function of \( \sqrt{D_k} \) in (28), which mainly comes from the fact that in general the estimation bias \( b_k \) is nonzero and may tend to zero if \( \gamma_k \to 0 \), as discussed in Section V-A. The additional term of \( \sqrt{D_k} \) makes our analysis much more challenging than that of the classical stochastic approximation [38], which assumes that an unbiased observation of gradient is always available.

Since \( \beta_k \gamma_k \) is vanishing, there should be \( 1 - A \beta_k \gamma_k > 0 \) when \( k \) is large enough. In this work we are interested in the bound of \( D_k \) when \( k \) is large. Introduce
\[ K_0 = \arg \min_{k \geq K_0, \beta_k \gamma_k < 1/A} k, \]
which implies that \( 1 - A \beta_k \gamma_k > 0 \) and \( k \geq K_c \) as \( k \to K_0 \). Lemma 4 provides us with the relation between \( D_{k+1} \) and \( D_k \). Our next goal is to find a vanishing upper bound of \( D_k \) using (28). When \( k \) is large, in other words, we aim to search a sequence \( U_{K_0}, U_k \) such that
\[ U_{k+1} \leq U_k \text{ and } D_k \leq U_k, \quad \forall k \geq K_0. \]

This type of analysis is usually performed by induction: consider a given expression of \( U_k \) and assume that \( D_k \leq U_k \), one needs to show that \( D_{k+1} \leq U_{k+1} \) by applying (28).

An important issue is then the proper choice of the form of the upper bound \( U_k \). Note that in the classical stochastic approximation algorithm described by (1) there exists only one step-size \( \beta_k \), and \( D_{k+1} \) is hence bounded by a linear function of \( D_k \). Our problem is much more complicated as we use two step-sizes \( \beta_k \) and \( \gamma_k \) with general form under Assumption 6. Moreover, the additional term \( \sqrt{D_k} \) presented in (28) also makes the analysis much more complicated. The following lemma presents an important property of \( U_k \).

**Lemma 5:** If there exists a decreasing sequence \( U_{K_0}, U_k \) such that \( D_{k+1} \leq U_{k+1} \) can be deduced from \( D_k \leq U_k \) and (28), then the following will be true
\[ U_k \geq \left( \frac{B}{2A} \beta_k + \frac{B}{2A} \gamma_k^2 + \frac{C}{A} \frac{\beta_k}{\gamma_k} \right)^2. \]  
(31)

**Proof:** See Appendix H.

Note that the lower bound of \( U_k \) will vanish only if \( \gamma_k \to 0 \) and \( \beta_k / \gamma_k \to 0 \). Such a bound means that, by using induction and applying (28), the convergence rate of \( D_k \) cannot be better than the decreasing speed of \( \gamma_k^2 \) and \( \beta_k / \gamma_k \). After the verification of the existence of bounded constants \( \theta \) such that \( D_k \leq \theta \gamma_k^2 \) or \( D_k \leq \theta \beta_k / \gamma_k \), we obtain the final results stated as follows.

**Theorem 3:** Define the following parameters:
\[ \chi_k = 1 - \frac{(\beta_k + 1)}{\beta_k \gamma_k}, \quad \epsilon_1 = \max_{k \geq K_0} \chi_k, \quad \epsilon_2 = \max_{k \geq K_0} \frac{\beta_k}{\gamma_k}, \]
\[ \varpi_k = 1 - \frac{\beta_k + 1}{\beta_k \gamma_k}, \quad \epsilon_3 = \max_{k \geq K_0} \varpi_k, \quad \epsilon_4 = \max_{k \geq K_0} \sqrt{\frac{\beta_k}{\gamma_k}}. \]  
(32)

If \( \chi_k < A \) for any \( k \geq K_0 \), then
\[ D_k \leq \theta^2 \gamma_k^2, \quad \forall k \geq K_0, \]  
with
\[ \theta \geq \max \left\{ \sqrt{\frac{D_{K_0}}{\gamma_{K_0}}}, \frac{B + \sqrt{B^2 + 4C \epsilon_2 (A - \epsilon_1)}}{2 (A - \epsilon_1)} \right\}. \]  
(35)

If \( \varpi_k < A \) for any \( k \geq K_0 \), then
\[ D_k \leq \theta^2 \beta_k, \quad \forall k \geq K_0, \]  
with
\[ \theta \geq \max \left\{ \sqrt{\frac{D_{K_0} \gamma_{K_0}}{C \epsilon_4^2}}, \frac{B \epsilon_4 + \sqrt{(B \epsilon_4)^2 + 4C (A - \epsilon_3)}}{2 (A - \epsilon_3)} \right\}. \]  
(37)

**Proof:** See Appendix I.

For general forms of vanishing \( \beta_k \) and \( \gamma_k \), Theorem 3 provides two upper bounds on the average divergence \( D_k^{(C)} \) and \( D_k^{(0)} \). The conditions \( \chi_k < A \) and \( \varpi_k < A \) can be checked easily by considering any fixed \( \beta_k \) and \( \gamma_k \). In the situation where both conditions are satisfied, we have
\[ D_k \leq \min \left\{ \theta^2 \beta_k, \theta^2 \gamma_k \right\}, \quad \forall k \geq K_0. \]

From Theorem 3, we can see that the decreasing order of \( D_k \) depends mainly on the step-sizes \( \beta_k \) and \( \gamma_k \). The incompleteness factor \( \theta \) has only an influence on the constant terms \( \vartheta \) and \( \varphi \), which are functions of the parameters \( A, B, \) and \( C \) defined by (29). The results of convergence rate are useful to properly choose the parameters of the algorithm. Intuitively, we need to make \( \gamma_k^2 \) or \( \beta_k / \gamma_k \) decrease as fast as possible, having the constant \( \epsilon_1, \epsilon_2, \epsilon_3, \) and \( \epsilon_4 \) as small as possible.

**B. Special Setting of Step-Sizes**

In this section, we consider the step-sizes as mentioned in Example 2 in Section IV-B:
\[ \beta_k = \beta_0 k^{-\nu_1} \quad \text{and} \quad \gamma_k = \gamma_0 k^{-\nu_2}. \]  
(38)

Recall that \( 0.5 < \nu_1 \leq 1 \) and \( 0 < \nu_2 \leq 1 - \nu_1 \) in order to meet the conditions in Assumption 6.

**Theorem 4:** Consider \( \beta_k \) and \( \gamma_k \) with forms given in (38), if \( \beta_0 \gamma_0 \geq \min \{ 2 \nu_2, \nu_1 - \nu_2 \} / \Omega \), then there exists \( \Omega < \infty \), such that
\[ D_k \leq \Omega k^{-\min\{2 \nu_2, \nu_1 - \nu_2 \}}, \quad \forall k \geq K_0. \]  
(39)

**Proof:** See Appendix J.
From Theorem 4, we get a simple function of $\nu_1$ and $\nu_2$ that determines the minimum convergence speed of $D_k$. It is easy to show
\[ \min \{2\nu_2, \nu_1 - \nu_2\} \leq 0.5 \]
with the equality holds when $\nu_1 = 0.75$ and $\nu_2 = 0.25$, which corresponds to the best choice of $\nu_1$ and $\nu_2$ to optimize the decreasing order of the upper bound of $D_k$. Theorem 4 also provides a sufficient condition on the parameters that should be satisfied in order to ensure the validation of the convergence rate, i.e., the value of $\beta_0 \gamma_0$ should be large enough.

With $\nu_1 = 0.75$ and $\nu_2 = 0.25$, we can further find the optimum values of $\beta_0$ and $\gamma_0$ that minimize the constant term of the upper bound of $D_k$. The result is shown in the following theorem.

**Theorem 5:** Suppose that Assumptions 1-7 hold and $a_k$ is updated by applying Algorithm 1, if the step-sizes are chosen as
\[ \beta_k = \frac{\alpha \Phi L_G}{\sqrt{2q\alpha \Phi \sigma_q^2 k^2}}, \quad \gamma_k = \frac{\sqrt{2\alpha \Phi}}{q\alpha \Phi \sigma_q^2 L_G} N^{-\frac{1}{2}}k^{-\frac{1}{2}}, \]
then we have, for any $k \geq K_0$,
\[ D_k \leq \max \left\{ (1 + K_0)^{-\frac{1}{2}} D_{K_0}, \frac{\sqrt{2q\alpha \Phi L_G}}{k^2 \sigma_q^2 q} N^2 \right\} k^{-\frac{1}{2}}. \]

**Proof:** See Appendix K.

Theorem 5 explicitly shows how each parameter affects the convergence rate of the proposed algorithm. It is worth mentioning that we do not conclude that $\beta_k$ and $\gamma_k$ in (40) are the optimal step-sizes, as they only minimize an upper bound of $D_k$. From the definition of $\alpha \Phi$ in (24), we find that $\alpha \Phi \propto N^{-\frac{1}{2}}$. Thus $D_k = O \left( N^3 k^{-\frac{3}{2}} \right)$. This result implies that the constant term of the convergence rate is also affected by the number of nodes in the network. The increase of the convergence rate with $N$ is at most of order $O \left( N^3 \right)$. It is worth mentioning, however, that the constant term of the upper bound may not be tight, as it includes several upper bounds of different parameters. We therefore provide numerical results in Section VII to illustrate the impact of the number of nodes on the convergence rate. Nevertheless, this aforementioned result implies that our algorithm would perform better in small to mid-sized networks.

1. **Optimality of the Optimization Error Rate:** It is worth mentioning that our result can be compared with the lower bound on the optimization error rate order derived in [12]–[14]. It is shown in [13] that the lower bound $O \left( K^{-0.5} \right)$ is actually the optimal DFSO optimization error rate for smooth and strongly concave functions. Note that the optimization error is defined as $F(a^*) - \mathbb{E}[F(\frac{1}{K} \sum_{k=1}^{K} a_k)]$. Recall that $\| \nabla F(a) - \nabla F(a') \| \leq L_G \| a - a' \|$, $\forall a, a' \in A$, by Assumption 2. This means that
\[ F(a^*) - F(a_k) \leq L_G \frac{1}{2} \| a_k - a^* \|^2, \]
\[ \mathbb{E} [ F(a^*) - F(a_k) ] \leq L_G \frac{1}{2} \mathbb{E} \left[ \| a_k - a^* \|^2 \right] \leq L_G \Omega \frac{1}{2} k^{-0.5} \]
according to Theorems 4 and 5 when $\beta_k \propto k^{-0.75}$ and $\gamma_k \propto k^{-0.25}$. By Jensen’s inequality, we get
\[ F(a^*) - \mathbb{E}[F(\frac{1}{K} \sum_{k=1}^{K} a_k)] \leq F(a^*) - \mathbb{E}[F(a_k) - \mathbb{E}[F(a_k)]] \leq \frac{L_G \Omega}{2} k^{-0.5} \]
We can see clearly that our achievable rate of optimization error is of order $O \left( K^{-0.5} \right)$, which coincides with the lower bound in [12]–[14].

This implies that our achieved optimization error rate is optimal, in the sense that the decay order of the optimization error with $K$ cannot be faster than the achieved one by our algorithm ($O \left( K^{-0.5} \right)$). This result is of interest since it implies that the optimal rate order obtained by centralized DFSO methods can also be achieved in the challenging distributed setting considered in this paper. Recall that in this paper the decision of each node is done independently from other nodes, and a node cannot be aware of the utilities of all other nodes.

**C. Verification**

In this section, we present a toy example to verify Theorem 4.

**Example 3:** Consider $N = 2$ and a simple function $f(a, S) = -s_1 a_1^2 - s_2 a_2^2 + a_1 a_2 + 4 \log(a_1) + 4 \log(a_2)$ with $a_1, a_2 \in [1, 4]$. Both $s_1$ and $s_2$ are realizations of a uniform distributed random variable $S$ which takes value in an interval $[0, 5]$. By taking the average, the objective function is
\[ F(a) = -a_1^2 - a_2^2 + a_1 a_2 + 4 \log(a_1) + 4 \log(a_2). \]

We can deduce the gradient and the Hessian matrix of $F$:
\[ \nabla F(a) = \left[ -2a_1 + 2a_2 + \frac{4}{a_1}, \quad -2a_2 + a_1 + \frac{1}{a_2} \right]^T; \]
\[ \nabla^2 F(a) = \begin{bmatrix}
-2 - 4a_1^{-2} & 1 \\
1 & -2 - 4a_2^{-2}
\end{bmatrix}. \]

We get that $F(a)$ is a strongly concave function with parameter $\kappa = 1.25$, since the matrix $\nabla^2 F(a) + \frac{\sigma^2}{2} I$ is semi-definite negative when $a_1, a_2 \in [1, 4]$. By solving $\nabla^2 F(a^*) = 0$, we obtain the optimal value $a^* = [2, 2]^T$. We set the perturbation $\Delta k$ as introduced in Example 2 (Section IV.B), with $\sigma^2 = 1$. Thus we have $A = 2a_1^{-2}a_2^{-2} = 2.5$ by (29).

Let $\beta_k = \beta_0 k^{-0.75}$ and $\gamma_k = k^{-0.25}$. By setting $\beta_0 \in \{0.1, 0.2, 0.5, 2\}$, we can verify the importance of the condition $\beta_0 \gamma_0 \geq 0.2$ is $\max \{2\nu_2, \nu_1 - \nu_2\}/A$ as presented in Theorem 4. Figure 3 shows the comparison results between the divergence $D_k$, averaged over 1000 simulations, and the bound $\Omega k^{-0.5}$ in (39). Note that we are mainly interested in the decreasing order of $D_k$ for $K > K_0$. Therefore, we set the constant term $\Omega = 2$ mainly to facilitate the visual comparison of different curves. We find that the curves of $D_k$ decrease not-less-slowly than the bound as $\beta_0 \in \{0.28, 0.5\}$. When $\beta_0 = 0.23 < 0.25$, the convergence rate cannot be guaranteed, which further justifies our claim in Theorem 4.
Fig. 3. Comparison of the theoretical upper bound $O(k^{-0.5})$ with the evolution of the average divergence $D_k$ (by averaging over 2000 simulations). $\beta_0 = 0.75$ and $\gamma_0 = k^{-0.25}$, with $\beta_0 \in \{0.1, 0.2, 2\}$.

Fig. 4. Comparison of the theoretical upper bound $O(k^{-0.3})$ with the evolution of the average divergence $D_k$ (by averaging over 2000 simulations). $\beta_k = \min(2\nu, \nu_1 - \nu_2)k^{-\nu_1}$ and $\gamma_k = k^{-\nu_2}$, with $(\nu_1, \nu_2) \in \{(0.85, 0.15), (0.7, 0.15), (0.65, 0.35), (0.5, 0.2)\}$.

Now we consider $(\nu_1, \nu_2) \in \{(0.85, 0.15), (0.7, 0.15), (0.65, 0.35), (0.5, 0.2)\}$, $\beta_0 = \min(2\nu, \nu_1 - \nu_2)/A$ and $\gamma_0 = 1$. Notice that all the four pairs of $(\nu_1, \nu_2)$ lead to the same decreasing order of the upper bound (39), as $\min(2\nu, \nu_1 - \nu_2) = 0.3$. The upper bound in this case is $O(k^{-0.3})$. The curves of $D_k$ (averaged over 2000 simulations) are compared with the upper bound $O(k^{-0.3})$ in Figure 4. Clearly, when the number of iterations is large enough, all the curves decrease with the same speed or even faster as compared with the bound, which again confirms our theoretical results.

VII. SIMULATION RESULTS

In this section, we first describe an application example that shows the limit of classical gradient-based methods. Then we present in Sections VII.B and VII.C some numerical results for this example to assess the performance of the proposed algorithm. In order to verify the convergence of our algorithm for high number of nodes, we present in Section VII.D another example and provide numerical results for high number of nodes.

A. Application Example

We consider the power allocation problem in a general network with $N$ transmitter-receiver links. A link here can be seen as a node in our system model as presented in Section III. For example,

- in a D2D network with $N$ transmitter-receiver pairs, each transmitter communicates with its associated receiver and different links interfere among each other, see Figure 5 (I).
- in a multi-cell network, each base station may serve multiple users. We focus on the downlink, i.e., a user is seen as a receiver and its associated base station is seen as a transmitter, see Figure 5 (II).

In these examples, $S_k$ represents the time-varying channel state of the network. More precisely, $S_k = \{s_{ij,k}\}_{i,j \in N, k \in N}$, in which each element $s_{ij,k} \geq 0$, denotes the power gain between transmitter $i$ and receiver $j$ at timeslot $k$.

There are various utility functions that can be considered in this case such as the total sum rate or the energy efficiency of the users [39]. In this section, we consider a widely used example in which objective function is the total sum rate of all users (i.e. sum-rate maximization problem). Suppose that each active transmitter $i$ controls its transmission power $p_{i,k}$, the Shannon achievable rate of the link is then given by

$$r_{i,k} = \log(1 + \frac{s_{ii,k}p_{i,k}}{\sigma^2 + \sum_{j \neq i} s_{ji,k}p_{j,k}}).$$

At each time-slot $k$, define the global utility, $y_k(p_k, S_k) = \sum_{i \in N}(\omega_1 r_{i,k} - \omega_2 p_{i,k})$, where $\omega_1, \omega_2 \in \mathbb{R}^+$ are constants and $\omega_2 p_{i,k}$ denotes the energy costs of the packet transmission of link $i$. However, $y_k$ is not concave with respect to $p_k$. Thus we consider a well known approximation of $r_{i,k}$ and some variable change to make the objective function concave. This transformation of the sum rate maximization problem is widely used in wireless networks and the details are therefore omitted for brevity. One can refer to [40] for more details. In fact, it is common to use the change of variable $p_{i,k} = e^{a_{i,k}}$ and to consider the approximation $y_k \approx f_k = \sum_{i \in N} u_i(a_k, S_k)$ with [40]

$$u_i(a_k, S_k) = \omega_1 \log\left(\frac{s_{ii,k}e^{a_{i,k}}}{\sigma^2 + \sum_{j \neq i} s_{ji,k}e^{a_{j,k}}}ight) - \omega_2 e^{a_{i,k}}.$$
We can write the partial derivative of the objective function as follows
\[
\tilde{g}_{i,k} = \frac{\partial f}{\partial a_{i,k}} = \omega_1 - \omega_1 \sum_{n \in \mathcal{N} \setminus \{i\}} \sigma_i^2 + \sum_{j \neq n} s_{j,n,k} e^{a_{j,k}} - \omega_2 e^{a_{i,k}},
\]
(45)

One can see that the aforementioned calculation requires the full knowledge of the cross-channel gain \(s_{n,k}, \forall n \in \mathcal{N} \setminus \{i\}\) and \(s_{j,n,k}, \forall j \in \mathcal{N} \setminus \{n\}\). All these channel information has to be estimated and exchanged by each node. Recall that the channels \(S_k\) are time varying which means that the computation of the derivatives, and hence the use of the gradient descent method, requires the estimation and exchanges of all these channel gains \(s_{n,k}\) at each time/iteration, which represents a huge burden for the network.

This example shows clearly the limit of the classical gradient-based methods and motivates the use of our solution that requires low information exchange at each time. Note that limiting the signaling overhead at each time is very critical in wireless networks since the amount of tolerated signaling information at each time is limited in such networks. It is worth mentioning that, apart from the example presented in this section, the solution proposed in this paper can also be applied to other types of problems such as beamforming, coordinated multipoint resource allocation (CoMP) [41], and so on.

In what follows, we apply our algorithm to solve the above optimization problem. The time varying channel \(h_{ij}\) between node \(i\) (transmitter) and node \(j\) (receiver) is generated using a Gaussian distribution with variance \(\sigma_i^2 = 1\) and \(\sigma_{ij} = 0.1, \forall i \neq j\). Notice that the channel gain is \(s_{ij} = |h_{ij}|^2\). Besides, we set \(\sigma_i^2 = 0.2, \omega_1 = 20\) and \(\omega_2 = 1\).

B. Simulations With 4 Nodes

In this section, we consider \(N = 4\) nodes. In all the simulations, the initial values of \(a_{0,i}, \forall i \in \mathcal{N}\) are generated uniformly in the interval \([\alpha_{i,\min}, \alpha_{i,\max}]\). In Algorithm 1, \(\Phi_{t,k}\) follows the symmetrical Bernoulli distribution, i.e.,
\[
P[\Phi_{i,k} = 1] = P[\Phi_{i,k} = -1] = 0.5, \forall k, i.
\]

We first compare our proposed algorithm with the sine perturbation based algorithm in [34]. We consider \(\alpha_{i,\min} = -5\) and \(\alpha_{i,\max} = 5\). The step-sizes are set as \(\beta_1 = 0.1k^{-0.75}\) and \(\gamma_k = \min \{2, 5k^{-0.25}\}\) in the proposed Algorithm 1. Notice that the sine perturbation based algorithm [34] is not easy to implement in practice as it is hard to choose all the parameters properly, especially when \(|\mathcal{N}|\) is large. Furthermore, in order to show the efficiency of our algorithm, we simulate also an ideal gradient-based algorithm (with step-size \(\beta_k = 2.5k^{-1}\)), in which the exact gradient calculated by (45) is known (recall that this gradient is costly to be obtained in practice as discussed in Section VII-A).

We have performed 500 independent simulations to obtain the average results shown in Figures 6 and 7. Figure 6 shows the utility function \(f(\alpha; S)\) versus the number of iterations. We find that our algorithm converges faster than the reference algorithm proposed in [34]. Figure 7 shows the evolution of the power (action) of the four nodes. Notice that the four curves representing the action of each node are close in average in each sub-figure, since we consider a model with symmetric parameters. We find the oscillation of the power (action) is more significant by applying the reference algorithm.

We then consider another scenario where we set \(\alpha_{1,\min} = 0\) and \(\alpha_{1,\max} = 5\), so that the objective function \(F\) is strongly concave. We perform 1000 independent simulations with \(p \in \{1, 0.5, 0.25, 0.1\}\). Recall that \(p\) defined in (10) represents the level of incompleteness. The step-sizes are chosen as \(\gamma_k = \min \{1.5, 15k^{-0.25}\}\) and \(\beta_k = \beta_0 k^{-0.75}\) with \(\beta_0 = \frac{1}{60(1-(1-p)^2)}\), which leads to \(\beta_k \gamma_k = \min \{2p_1, \nu_1 - \nu_2\}\). The results are shown in Figure 8. We can see that after sufficient number of iterations, the different curves are approximately parallel and decrease with the same speed as the theoretical bound \(O(k^{-0.5})\) obtained in Theorem 4. The results also justify that the decreasing order of \(D_k\) is not affected by the value of \(p\).

1 The reference algorithm is quite sensitive to the parameters, the presented results are the best that we have found so far.
2 The strong concavity comes from the fact that \(\sum \log (\sinh a_i)\) is concave with respect to \(a_i\) and the exponential function \(-\exp(\alpha_{i,k})\) is \(\exp(\alpha_{i,\min}) - \text{concave for any } \alpha_{i,k} \geq \alpha_{i,\min}\).
negligible. The following example is therefore considered. In fact, in the previous example, the interference in high number of nodes, we consider in this subsection another example similar to those in Figure 8. The algorithm converges slower and the shape of the curves in Figure 9 is similar to those in Figure 8. The algorithm converges slower as the number of nodes increases, yet the influence of the incompleteness of the received local utilities is less important.

C. Simulations With 10 Nodes

In this section, we consider a more challenging case with \( N = 10 \). We choose \( \gamma_k = \min\{1, 12k^{-0.25}\} \) and \( \beta_k = \frac{k^{-0.75}}{48(1-(1-p)^p)} \). The results are presented in Figure 9 with \( p \in \{1, 0.5, 0.25, 0.1\} \). Note that we have also plotted on this figure a line representing the optimum value of the average utility function. The shape of the curves in Figure 9 is similar to those in Figure 8. The algorithm converges slower as the number of nodes increases, yet the influence of the incompleteness of the received local utilities is less important.

D. Simulations With High Number of Nodes

In order to assess the performance of our algorithm for high number of nodes, we consider in this subsection another example. In fact, in the previous example, the interference in the network increases with the number of nodes \( N \), and hence for high values of \( N \) the utility function of each node becomes negligible. The following example is therefore considered.

Consider \( a = [a_1, ..., a_N] \in [1, 4]^N \) and a simple function
\[
f(a; S) = \sum_{i=1}^{N} \left( 4 \log(a_i) - s_i a_i^2 \right) + \frac{1}{2(N-1)} \sum_{i=1}^{N} \sum_{j \neq i} a_i a_j\]
where \( s_i \) for all \( i \in \{1, ..., N\} \) are realizations of a uniform distributed random variable \( S \) taking values in an interval \([0.5, 1.5]\). By taking the average with respect to \( S \), we have the objective function
\[
F(a) = \sum_{i=1}^{N} \left( 4 \log(a_i) - a_i^2 \right) + \frac{1}{2(N-1)} \sum_{i=1}^{N} \sum_{j \neq i} a_i a_j.
\]
It is straightforward to verify that \( F(a) \) takes its maximum value when \( a = a^* = [2, ..., 2] \).

To perform the proposed algorithm, the step-sizes are set as \( \beta_k = \frac{100}{N} k^{-0.75} \) and \( \gamma_k = \min\{0.5, 10k^{-0.25}\} \). We consider the dimension of the problem as \( N \in \{20, 100, 500\} \). Figure 10 presents the evolution of \( D_k/N \) for multiple values of \( N \). Results show that the algorithm takes more time to converge when \( N \) is high. Comparing the curves in Figure 10, we find that the value of \( D_k/N \) approximately increases linearly with \( N \). This means that, in the considered example, the increase of \( D_k \) with respect to \( N \) is less that the theoretical upper bound derived in Theorem 5 \( D_k = O\left(N^3 k^{-\frac{1}{2}}\right) \). On the other hand, the upper bound is valid for a more general class of smooth and strongly concave function, while the objective function considered here is relatively simple. The obtained results here suggest that our proposed algorithm performs better in small to mid-sized networks.

VIII. Conclusion

In this paper we have addressed a challenging distributed derivative-free stochastic optimization (DFSO), for which classical gradient-based methods cannot be applied. The objective is to optimize the average total sum of the local utilities of the nodes, under the assumptions that each node knows at each time a numerical value of its local utility and part of the local utilities of other nodes. We developed a distributed algorithm in which, each node performs an estimation of the global objective function using the incomplete knowledge of the local utilities of other nodes. Each node updates then its action independently from other nodes, by using an appropriate stochastic perturbation of its estimated value of the objective function. We proved that the proposed algorithm converges almost surely to the optimum, under the assumption that the objective function is smooth and strongly
concave, the convergence rate of the algorithm is also derived. We have shown that, after sufficient number of iterations $K > K_0$, it scales as $O(K^{-0.5})$, which is the optimal convergence rate for smooth and strongly concave functions. Numerical results are also provided to corroborate our claims.

**APPENDIX**

A. Proof of Lemma 1

We start with the conditional expectation, based on (11), we have

$$
\mathbb{E}_{I_x,k} \left[ \bar{f}_{i,k} (a; S_k, \eta_k, I_x,k) \mid [I_x,k] = n \right] = \mathbb{E}_{I_x,k} \left[ \bar{u}_{j,k} [I_x,k] = n \right], \text{ if } n \neq 0,
$$

and

$$
0, \text{ if } n = 0.
$$

(46)

Denote $U^{(n)}$ as a collection of all possible sets $I_x,k$ such that $[I_x,k] = n$, e.g., $U^{(1)} = \{\{1\}, \{2\}, \ldots, \{N\} \}$. Since each node has an equal probability to be involved in $I_x,k$, the sets in $U^{(n)}$ are also equiprobable, i.e.,

$$
P \left[ I_x,k = I \mid [I_x,k] = n \right] = \frac{1}{(N-1)}, \forall I \in U^{(n)},
$$

note that the cardinal of $U^{(n)}$ is $(N-1)$. We evaluate

$$
\mathbb{E}_{I_x,k} \left[ \bar{u}_{j,k} \mid [I_x,k] = n \right] = \frac{1}{N-1} \sum_{j \in I_x,k} \bar{u}_{j,k} = \frac{1}{N-1} \sum_{j \in I_x,k} \bar{u}_{j,k}.
$$

(47)

Combine (46) and (47), for any $n \in \{1, \ldots, N-1\}$, we have

$$
\mathbb{E}_{I_x,k} \left[ \bar{f}_{i,k} \mid [I_x,k] = n \right] = \sum_{j \in N} \bar{u}_{j,k}.
$$

(48)

According to the basic rule of expectation, we have

$$
\mathbb{E}_{I_x,k} \left[ \bar{f}_{i,k} \right] = \sum_{n=0}^{N-1} \mathbb{E}_{I_x,k} \left[ \bar{f}_{i,k} \mid [I_x,k] = n \right] \mathbb{P} \left[ [I_x,k] = n \right]
$$

$$
= \sum_{n=0}^{N-1} \mathbb{P} \left[ [I_x,k] = n \right] \sum_{j \in N} \bar{u}_{j,k}
$$

$$
= (1 - \mathbb{P} \left[ [I_x,k] = 0 \right]) \sum_{j \in N} \bar{u}_{j,k}
$$

$$
= q \sum_{j \in N} \bar{u}_{j,k},
$$

where $q$ is defined in (13). Since $\bar{u}_{j,k} = u_j (a; S_k) + \eta_{j,k}$ and $\mathbb{E} [\eta_{i,k}] = 0$ by Assumption 4, we have $\mathbb{E}_{I_x,k,\eta_k} \left[ \bar{f}_{i,k} \right] = q \sum_{j \in N} u_j (a; S_k) = q f (a; S_k)$, which concludes the proof.

B. Proof of Theorem 1

In this proof, we mainly need to find the relation between $\bar{f}_k$ (the expected estimation of the gradient) and $G(a_k) = \nabla F(a_k)$ (the actual gradient of the objective function), so that the upper bound of $\|b_k\|$ can be derived.

From the definition of $\bar{f}_{i,k}$, we have

$$
\bar{f}_{i,k} = \mathbb{E}_{S,\Phi,\eta} \left[ \Phi_{i,k} f_{i,k} \right] = q \mathbb{E}_{S,\Phi} \left[ \Phi_{i,k} \mathbb{E}_S [f(a_k + \gamma_k \Phi_k; S_k)] \right] = q \mathbb{E}_{S,\Phi} \left[ \Phi_{i,k} F(a_k + \gamma_k \Phi_k) \right],
$$

(49)

note that the $\mathbb{E}_{S,\Phi} \left[ \bar{f}_{i,k} \right] = q f (a_k + \gamma_k \Phi_k; S_k)$ by Lemma 1 and $F$ is the expected value of $f$ by definition.

Based on Taylor’s theorem and mean-valued theorem, there exists $\bar{a}_k$ locating between $a_k$ and $a_k + \gamma_k \Phi_k$ such that

$$
F(a_k + \gamma_k \Phi_k) = F(a_k) + \sum_{j \in N} \gamma_k \Phi_{j,k} G_j (a_k),
$$

+ \sum_{j_1, j_2 \in N} \frac{\gamma_k^2}{2} \Phi_{j_1,k} \Phi_{j_2,k} \frac{\partial^2}{\partial a_{j_1} \partial a_{j_2}} F (\bar{a}_k).
$$

(50)

Substituting (50) into (49), we have, $\forall I \in N$,

$$
\mathbb{E}_{I_x,k} = q F (a_k) \mathbb{E}_{\Phi} \left[ \Phi_{i,k} f_{i,k} \right] + q \gamma_k \sum_{j \in N} G_j (a_k) \mathbb{E}_{\Phi} \left[ \Phi_{i,k} \Phi_{j,k} \right] + q \gamma_k \sum_{j_1, j_2 \in N} \frac{\gamma_k^2}{2} \mathbb{E}_{\Phi} \left[ \Phi_{j_1,k} \Phi_{j_2,k} \frac{\partial^2}{\partial a_{j_1} \partial a_{j_2}} F (\bar{a}_k) \right] = \sigma_k^2 q \gamma_k (G_i (a_k) + b_{i,k}),
$$

(51)

from which we get the bias term

$$
b_{i,k} = \sum_{j_1, j_2 \in N} \frac{\gamma_k}{2} \mathbb{E}_{\Phi} \left[ \Phi_{j_1,k} \Phi_{j_2,k} \frac{\partial^2}{\partial a_{j_1} \partial a_{j_2}} F (\bar{a}_k) \right].
$$

(52)

Note that (51) is obtained by considering Assumption 6 which states that $\mathbb{E}_{\Phi} \left[ \Phi_{i,k} \right] = 0$, $\mathbb{E}_{\Phi} \left[ \Phi_{i,k}^2 \right] = \sigma_k^2$, and $\mathbb{E}_{\Phi} \left[ \Phi_{i,k} \Phi_{j,k} \right] = 0, \forall i \neq j$. It is worth mentioning that in general $\frac{\partial^2}{\partial a_{j_1} \partial a_{j_2}} F (\bar{a}_k)$ is also an implicit function of $\Phi_k$, thus the closed form expression of $b_{i,k}$ may be complicated to evaluate. Nevertheless, we are interested in the upper bound of $b_{i,k}$.

Since $F$ is concave and $L_G$-smooth, we have

$$
F(a_k + \gamma_k \Phi_k) < F(a_k) + \sum_{j \in N} \gamma_k \Phi_{j,k} G_j (a_k);
$$

(53)

$$
F(a_k + \gamma_k \Phi_k) \geq F(a_k) + \sum_{j \in N} \gamma_k \Phi_{j,k} G_j (a_k) - \frac{L_G}{2} \|\gamma_k \Phi_k\|^2.
$$

(54)

By comparing (53)-(54) with (50), we get

$$
-\frac{L_G}{2} \|\gamma_k \Phi_k\|^2 \leq \sum_{j_1, j_2 \in N} \frac{\gamma_k^2}{2} \Phi_{j_1,k} \Phi_{j_2,k} \frac{\partial^2}{\partial a_{j_1} \partial a_{j_2}} F (\bar{a}_k) < 0
$$

which means that

$$
\sum_{j_1, j_2 \in N} \Phi_{j_1,k} \Phi_{j_2,k} \frac{\partial^2}{\partial a_{j_1} \partial a_{j_2}} F (\bar{a}_k) \leq L_G \|\Phi_k\|^2.
$$

(55)
From (52) and (55), we have

$$\|b_{i,k}\| = \frac{\gamma_k}{\sigma_F} \frac{\gamma_k}{2\sigma_F} \mathbb{E}_F \left[ \Phi_{i,k} \sum_{j_1,j_2 \in N} \Phi_{j_1,k} \Phi_{j_2,k} \frac{\partial^2}{\partial a_{j_1} \partial a_{j_2}} F(\tilde{a}_{j_1}) \right]$$

$$\leq \frac{\gamma_k}{\sigma_F} \frac{\gamma_k}{2\sigma_F} \mathbb{E}_F \left[ |\Phi_{i,k}| \left| \sum_{j_1,j_2 \in N} \Phi_{j_1,k} \Phi_{j_2,k} \frac{\partial^2}{\partial a_{j_1} \partial a_{j_2}} F(\tilde{a}_{j_2}) \right| \right]$$

$$\leq \frac{\gamma_k}{\sigma_F} \frac{\gamma_k}{2\sigma_F} \mathbb{E}_F \left[ \alpha_{2} \left| \Phi_{i,k} \right|^2 \right] = \frac{\gamma_k}{\sigma_F} \frac{\gamma_k}{2\sigma_F} \mathbb{E}_F \left[ \sum_{i \in N} \Phi_{i,k}^2 \right]$$

$$= \frac{\gamma_k}{\sigma_F} \frac{\gamma_k}{2\sigma_F} N \sigma_F^2 = \frac{\alpha_{2}}{2} \mathbb{E}_F \left[ \sum_{i \in N} \Phi_{i,k}^2 \right].$$

Then (21) can be proved as $$\|b_{i,k}\| = \sqrt{\sum_{i=1}^{N} |b_{i,k}|^2}.$$ by the assumption that $$\mathbb{E} \left[ \eta_{j,k}^2 \right] = \sigma_{\eta}^2.$$ Then we consider

$$\mathbb{E}_{\mathcal{I}_{i,k}} \left[ \left( u_{i,k} + \frac{N-1}{n} \sum_{j \in \mathcal{I}_{i,k}} u_j \right)^2 \right] = \mathbb{E}_{\mathcal{I}_{i,k}} \left[ \left( u_{i,k} + \frac{N-1}{n} \sum_{j \in \mathcal{I}_{i,k}} u_j \right)^2 \right]$$

$$\leq \mathbb{E}_{\mathcal{I}_{i,k}} \left[ \left( u_{i,k} + \frac{N-1}{n} \sum_{j \in \mathcal{I}_{i,k}} u_j \right)^2 \right]$$

$$= (n+1) \left( \frac{2N-1}{n} \right) \mathbb{E}_{\mathcal{I}_{i,k}} \left[ \sum_{j \in \mathcal{I}_{i,k}} u_j^2 \right]$$

$$= (n+1) \left( \frac{2N-1}{n} \right) \sum_{j \in \mathcal{I}_{i,k}} u_j^2$$

$$= (n+1) \left( \frac{2N-1}{n} \right) \sum_{j \in \mathcal{I}_{i,k}} u_j^2.$$ (58)

where (a) is by the inequality $\sum_{i=1}^{m} x_i/m \leq \sqrt{\sum_{i=1}^{m} x_i^2/m}$ for all $x_i \in \mathbb{R}$ and (b) can be proved using the same way as (47).

Since $\tilde{g}_{i,k} = 0$ as $n = 0$ by definition, based on (57) and (58), we have

$$\mathbb{E}_{\mathcal{I}_{i,k}} \left[ \left( \tilde{g}_{i,k} \right)^2 \right]$$

$$= \sum_{n=1}^{N-1} \mathbb{P} \left[ |\mathcal{I}_{i,k}| = n \right] \mathbb{E}_{\mathcal{I}_{i,k}} \left[ \left( \tilde{g}_{i,k} \right)^2 \right]$$

$$\mathbb{P} \left[ |\mathcal{I}_{i,k}| = n \right] = \mathbb{P} \left[ |\mathcal{I}_{i,k}| \neq 0 \right] = q,$$

$$\mathbb{P} \left[ |\mathcal{I}_{i,k}| = n \right] = \mathbb{P} \left[ |\mathcal{I}_{i,k}| \neq 0 \right] = q,$$

$$\mathbb{P} \left[ |\mathcal{I}_{i,k}| = n \right] = \mathbb{P} \left[ |\mathcal{I}_{i,k}| \neq 0 \right] = q,$$

$$\mathbb{P} \left[ |\mathcal{I}_{i,k}| = n \right] = \mathbb{P} \left[ |\mathcal{I}_{i,k}| \neq 0 \right] = q,$$

$$\mathbb{P} \left[ |\mathcal{I}_{i,k}| = n \right] = \mathbb{P} \left[ |\mathcal{I}_{i,k}| \neq 0 \right] = q,$$

$$\mathbb{P} \left[ |\mathcal{I}_{i,k}| = n \right] = \mathbb{P} \left[ |\mathcal{I}_{i,k}| \neq 0 \right] = q,$$

$$\mathbb{P} \left[ |\mathcal{I}_{i,k}| = n \right] = \mathbb{P} \left[ |\mathcal{I}_{i,k}| \neq 0 \right] = q,$$

$$\mathbb{P} \left[ |\mathcal{I}_{i,k}| = n \right] = \mathbb{P} \left[ |\mathcal{I}_{i,k}| \neq 0 \right] = q,$$

$$\mathbb{P} \left[ |\mathcal{I}_{i,k}| = n \right] = \mathbb{P} \left[ |\mathcal{I}_{i,k}| \neq 0 \right] = q,$$

$$\mathbb{P} \left[ |\mathcal{I}_{i,k}| = n \right] = \mathbb{P} \left[ |\mathcal{I}_{i,k}| \neq 0 \right] = q,$$

$$\mathbb{P} \left[ |\mathcal{I}_{i,k}| = n \right] = \mathbb{P} \left[ |\mathcal{I}_{i,k}| \neq 0 \right] = q,$$

$$\mathbb{P} \left[ |\mathcal{I}_{i,k}| = n \right] = \mathbb{P} \left[ |\mathcal{I}_{i,k}| \neq 0 \right] = q,$$

$$\mathbb{P} \left[ |\mathcal{I}_{i,k}| = n \right] = \mathbb{P} \left[ |\mathcal{I}_{i,k}| \neq 0 \right] = q,$$

$$\mathbb{P} \left[ |\mathcal{I}_{i,k}| = n \right] = \mathbb{P} \left[ |\mathcal{I}_{i,k}| \neq 0 \right] = q,$$

$$\mathbb{P} \left[ |\mathcal{I}_{i,k}| = n \right] = \mathbb{P} \left[ |\mathcal{I}_{i,k}| \neq 0 \right] = q,$$

$$\mathbb{P} \left[ |\mathcal{I}_{i,k}| = n \right] = \mathbb{P} \left[ |\mathcal{I}_{i,k}| \neq 0 \right] = q,$$

$$\mathbb{P} \left[ |\mathcal{I}_{i,k}| = n \right] = \mathbb{P} \left[ |\mathcal{I}_{i,k}| \neq 0 \right] = q,$$

$$\mathbb{P} \left[ |\mathcal{I}_{i,k}| = n \right] = \mathbb{P} \left[ |\mathcal{I}_{i,k}| \neq 0 \right] = q,$$

$$\mathbb{P} \left[ |\mathcal{I}_{i,k}| = n \right] = \mathbb{P} \left[ |\mathcal{I}_{i,k}| \neq 0 \right] = q,$$

$$\mathbb{P} \left[ |\mathcal{I}_{i,k}| = n \right] = \mathbb{P} \left[ |\mathcal{I}_{i,k}| \neq 0 \right] = q,$$

$$\mathbb{P} \left[ |\mathcal{I}_{i,k}| = n \right] = \mathbb{P} \left[ |\mathcal{I}_{i,k}| \neq 0 \right] = q,$$

$$\mathbb{P} \left[ |\mathcal{I}_{i,k}| = n \right] = \mathbb{P} \left[ |\mathcal{I}_{i,k}| \neq 0 \right] = q.$$
which can be proved by
\[
\sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N} \setminus \{i\}} u_{j,k}^2 = (N-1) \sum_{i \in \mathcal{N}} u_{i,k}^2
\]
and recall that \( \mathbb{E} [\eta_{i,k}^2] = \sigma_i^2 \) (Assumption 4), \( \mathbb{E} [\Phi_i^2] = \sigma_i^2 \) and \( |\phi_{i,k}| \leq \alpha_\phi \) (Assumption 6). Lemma 2 is then proved.

E. Proof of Lemma 3
The proof of Lemma 3 is mainly by the application of Doob’s martingale inequality [42].

We first need to show that the sequence \( \{K' = K \beta_k e_k \}_{K' \geq K} \) is martingale, which is straightforward as \( \hat{g}_k \) and \( \hat{g}_{k'} \) are independent if \( k \neq k' \) and \( \mathbb{E}_{\mathcal{N}, \eta, \mathcal{I}} [e_k] = \mathbb{E}_{\mathcal{N}, \eta, \mathcal{I}} [\hat{g}_k - \mathbb{E}_{\mathcal{N}, \eta, \mathcal{I}} [\hat{g}_k]] = 0 \).

Then, by Doob’s martingale inequality, for any positive constant \( \rho \), we have
\[
\mathbb{P} \left[ \sup_{K' \geq K} \left| \sum_{k=K}^{K'} \beta_k e_k \right| \geq \rho \right] \\
\leq \frac{1}{\rho^2} \mathbb{E}_{\mathcal{N}, \eta, \mathcal{I}} \left[ \left( \sum_{k=K}^{K'} \beta_k e_k \right)^2 \right] \\
= \frac{1}{\rho^2} \mathbb{E}_{\mathcal{N}, \eta, \mathcal{I}} \left[ \left( \sum_{k=K}^{K'} \beta_k e_k \right)^2 \right] \\
= \frac{1}{\rho^2} \mathbb{E}_{\mathcal{N}, \eta, \mathcal{I}} \left[ \left( \sum_{k=K}^{K'} \beta_k e_k \right)^2 \right] \\
\leq \frac{1}{\rho^2} \sum_{k=K}^{\infty} \mathbb{E}_{\mathcal{N}, \eta, \mathcal{I}} \left[ \beta_k^2 \| \hat{g}_k - \mathbb{E}_{\mathcal{N}, \eta, \mathcal{I}} [\hat{g}_k] \|^2 \right] \\
= \frac{1}{\rho^2} \beta_k^2 \left( \sum_{k=K}^{\infty} \mathbb{E}_{\mathcal{N}, \eta, \mathcal{I}} [\hat{g}_k] - \sum_{k=K}^{\infty} \mathbb{E}_{\mathcal{N}, \eta, \mathcal{I}} [\hat{g}_k] \right)^2 \\
\leq \frac{1}{\rho^2} \sum_{k=K}^{\infty} \beta_k^2 \mathbb{E}_{\mathcal{N}, \eta, \mathcal{I}} [\hat{g}_k] \leq \frac{\alpha_\phi^2}{\rho^2} \sum_{k=K}^{\infty} \beta_k \quad (61)
\]

where (a) holds as \( \mathbb{E} [e_k^T e_{k'}] = 0 \) for any \( k \neq k' \) and (b) is by Lemma 2. Since \( \lim_{K \to \infty} \sum_{k=K}^{\infty} \beta_k^2 = 0 \) by Assumption 6 and \( M \) is bounded, we have \( \sum_{k=K}^{\infty} \beta_k \) is vanishing for any bounded constant \( \rho \). Therefore, the probability that \( \| \sum_{k=K}^{K'} \beta_k e_k \| \geq \rho \) is also vanishing, which concludes the proof.

F. Proof of Theorem 2
In this proof, we start with the evolution of the divergence \( d_k \) as defined in (18), then we show that \( d_k \) should be vanishing almost surely by applying Theorem 1 and Lemma 3.

We learn first the property of the projection (15). Define \( C_{i,k} = [\alpha_i + \alpha_\phi \gamma_k, \alpha_i - \alpha_\phi \gamma_k] \) for any \( i \in \mathcal{N} \) and \( C_k = C_{1,k} \times \cdots \times C_{N,k} \). Obviously, \( a_{k+1} \in C_{k+1} \) by (15). Since \( \gamma_k \) is a decreasing sequence, we find that \( C_k \subseteq C_{k+1} \). Furthermore, we have \( \lim_{k \to \infty} C_k = \overline{[\alpha_{i,min} + \alpha_\phi \gamma_1, \alpha_{i,max} - \alpha_\phi \gamma_1]} \), as \( \lim_{k \to \infty} \gamma_k = 0 \). Hence, there exists \( K_c \in \mathbb{N} \), such that \( a^* \in C_{k+1} \) for any \( k \geq K_c \). Note that \( K_c = 1 \) if \( a^* \in [\alpha_{i,min} + \alpha_\phi \gamma_1, \alpha_{i,max} - \alpha_\phi \gamma_1] \), \( \forall i \in \mathcal{N} \). It is straightforward to prove that
\[
\| a_{k+1} - a^* \| \leq \| \tilde{a}_{k+1} - a^* \|, \forall k \geq K_c,
\]
meaning that the projection makes \( a_{k+1} \) closer to \( a^* \) in the Euclidean space when \( a_{k+1} \in C_{k+1} \) and \( a^* \in C_{k+1} \).

Considering the divergence as defined in (18), we have, \( \forall k \geq K_c \)
\[
d_{k+1} = \| a_{k+1} - a^* \|^2 \geq \| \tilde{a}_{k+1} - a^* \|^2 = \| a_k + \beta_k \hat{g}_k - a^* \|^2 = d_k + 2 \beta_k \tilde{g}_k a_k - a^* \|^2 + 2 \beta_k (a_k - a^*)^T \cdot \hat{g}_k.
\]

We sum both sides of (62) to obtain
\[
d_{k+1} = d_{k+1} + \sum_{k=Kc}^{K} \left( \beta_k^2 \| \hat{g}_k \|^2 + 2 \beta_k (a_k - a^*)^T \cdot \hat{g}_k \right).
\]

Considering (22), we can rewrite (63) as
\[
d_{k+1} = d_{k+1} + \sum_{k=Kc}^{K} \left( \beta_k^2 \| \hat{g}_k \|^2 + 2 \beta_k (a_k - a^*)^T \cdot \hat{g}_k \right)
+ 2 \sigma_q^2 \sum_{k=Kc}^{K} \beta_k \gamma_k (a_k - a^*)^T \cdot (\nabla F (a_k) + b_k).
\]

We have \( \lim_{K \to \infty} \sum_{k=Kc}^{K} \beta_k e_k \| \| < \| a_k - a^* \| < \infty \) a.s. from Lemma 3, as well as \( \| a_k - a^* \| < \infty \). Thus
\[
\lim_{K \to \infty} \sum_{k=Kc}^{K} \beta_k (a_k - a^*)^T \cdot e_k < \infty, \quad \text{a.s.}
\]

From Lemma 2, we have \( \| \hat{g}_k \|^2 < \infty \) almost surely. With \( \sum_{k=Kc}^{K} \beta_k^2 < \infty \), the following holds almost surely as well,
\[
\lim_{K \to \infty} \sum_{k=Kc}^{K} \beta_k^2 \| \hat{g}_k \|^2 < \infty.
\]

From the above equations (64)-(66), we conclude that there exists \( W < \infty \) such that \( d_{K+1} \leq W + 2 \sigma_q^2 \varepsilon_{z_k} \), with
\[
z_k = \sum_{k=Kc}^{K} \beta_k \gamma_k (a_k - a^*)^T \cdot (\nabla F (a_k) + b_k).
\]

Since \( d_{K+1} \geq 0 \) by definition, we have \( z_k > -\infty \).

By recheck (21), we have that \( \| b_k \| \) is vanishing with \( \gamma_k \). Hence, we can say that for an arbitrary small positive value \( \varepsilon_b \), there exists \( K_b \) such that,
\[
\| \nabla F (a_k) + b_k \| \geq (1 - \varepsilon_b) \| \nabla F (a_k) \|, \forall k \geq K_b,
\]
which also implies that \( \| a_k - a^* \|^2 \cdot (\nabla F (a_k) + b_k) \leq 0, \forall \| a_k - a^* \| \) by the concavity of \( F \) in (8). Therefore \( 0 \leq d_{K+1} < \infty \) for any large \( K \) and \( \lim_{K \to \infty} d_{K+1} = d \) exists.
The following steps of the proof is similar to the classical proof in [43].

Assume that: H1) $\overline{d} > 0$, i.e., $a_k$ does not converge to $a^*$, then for any $\varepsilon_h > 0$, there exists $K_h$ such that
\[
(a_k - a^*)^T \cdot \nabla F(a_k) < -\varepsilon_h, \forall k \geq K_h.
\]
(69)

From (67) and (68), we get that $\forall k \geq K_m = \max \{K_h, K_b, K_c\}$,
\[
(a_k - a^*)^T (\nabla F(a_k) + b_k) < -\varepsilon_h (1 - \varepsilon_h),
\]
(70)

which leads to
\[
\lim_{K \to \infty} \sum_{k=K_m}^K \beta_k \gamma_k (a_k - a^*)^T (\nabla F(a_k) + b_k) < -\varepsilon_h (1 - \varepsilon_h) \lim_{K \to \infty} \sum_{k=K_m}^K \beta_k \gamma_k < -\infty,
\]
(71)
as $\sum \beta_k \gamma_k$ diverges by Assumption 6.

We find that $z_k < -\infty$ and $d_{K+1} < -\infty$. However $d_{K+1}$ should be positive by definition. Therefore, the hypothesis H1 cannot be true, and there should be therefore $\lim_{k \to \infty} d_k = 0$, $\lim_{k \to \infty} \nabla F(a_k) = 0$, and $\lim_{k \to \infty} a_k = a^*$ almost surely, which concludes the proof.

G. Proof of Lemma 4

We investigate the relation between two successive values of the average divergence. In Algorithm 1, we have
\[
D_{k+1} = E \left[ \left\| a_{k+1} - a^* \right\|^2 \right] = E \left[ \left\| a_k + \beta_k \tilde{g}_k - a^* \right\|^2 \right] = D_k + \beta_k^2 E \left[ \left\| \tilde{g}_k \right\|^2 \right] + 2 \beta_k E \left[ (a_k - a^*)^T \tilde{g}_k \right].
\]
(72)

We can evaluate
\[
E \left[ (a_k - a^*)^T \tilde{g}_k \right] = \sigma_{\tilde{g}}^2 q_k \gamma_k E \left[ (a_k - a^*)^T \nabla F(a_k) + b_k \right],
\]
(73)

which can be obtained by using (22) and $E[e_k] = 0$. We have
\[
(a_k - a^*)^T b_k \leq \sum_{i=1}^N \left| a_{i,k} - a_{i}^* \right| \left| b_{i,k} \right|
\]
\[
\leq \frac{1}{2} \sigma_{\tilde{g}} L_G N \gamma_k \sum_{i=1}^N \left| a_{i,k} - a_{i}^* \right|
\]
\[
\leq \frac{1}{2} \sigma_{\tilde{g}} L_G N \gamma_k \sqrt{N \sum_{i=1}^N (a_{i,k} - a_{i}^*)^2}
\]
\[
= \frac{1}{2} \gamma_k \sigma_{\tilde{g}} L_G \gamma_k \left\| a_k - a^* \right\|.
\]
(74)

From (73), (74), and the strong concavity (27), we have
\[
E \left[ (a_k - a^*)^T \tilde{g}_k \right] \leq \sigma_{\tilde{g}}^2 q_k \gamma_k E \left[ -\kappa \left\| a_k - a^* \right\|^2 + \frac{\alpha \phi}{2} L_G N \frac{\gamma_k^2}{\beta_k} \left\| a_k - a^* \right\| \right]
\]
\[
= -\sigma_{\tilde{g}}^2 q_k \gamma_k E \left[ \left\| a_k - a^* \right\|^2 \right] + \frac{1}{2} \alpha \phi \sigma_{\tilde{g}}^2 q_k L_G \gamma_k \frac{\gamma_k^2}{\beta_k} \sqrt{E \left[ \left\| a_k - a^* \right\| \right]^2}
\]
\[
\leq -\sigma_{\tilde{g}}^2 q_k \gamma_k \beta_k d_k + \frac{1}{2} \alpha \phi \sigma_{\tilde{g}}^2 q_k L_G \gamma_k \frac{\gamma_k^2}{\beta_k} \sqrt{D_k}.
\]
(75)

By combining (72), (75) and the fact that $E[\left\| \tilde{g}_k \right\|^2] \leq \sigma_{\tilde{g}}^2$ from Lemma 2, we obtain
\[
D_{k+1} = (1 - 2 \sigma_{\tilde{g}}^2 q_k \beta_k \gamma_k) D_k + \alpha \phi \sigma_{\tilde{g}}^2 \frac{\gamma_k^2}{\beta_k} \frac{\gamma_k^2}{\beta_k} \sqrt{D_k}.
\]
(76)

Lemma 4 is then proved.

H. Proof of Lemma 5

From (28) and $D_k \leq U_k$, we get
\[
D_{k+1} \leq (1 - A \beta_k) U_k + B \beta_k \gamma_k \sqrt{U_k} + C \beta_k^2,
\]
because $1 - A \beta_k \gamma_k > 0$ when $k \geq K_0$. In order to perform the induction, we need to have
\[
(1 - A \beta_k \gamma_k) U_k + B \beta_k \gamma_k \sqrt{U_k} + C \beta_k^2 \leq U_{k+1} \leq U_k,
\]
which leads to
\[
A \gamma_k U_k - B \gamma_k \sqrt{U_k} - C \beta_k \geq 0.
\]
(77)

By solving (77), we obtain (31) as $\sqrt{U_k} > 0$.

I. Proof of Theorem 3

We start with the proof of (34), which is realized by induction.

It is straightforward to get $D_{K_0} \leq \vartheta_2 \gamma_{K_0}^2$ according to the definition of $\vartheta$. We mainly need to verify that $D_k \leq \vartheta^2 \gamma_k^2$ leads to $D_{k+1} \leq \vartheta^2 \gamma_{k+1}^2$ for any $k \geq K_0$.

Suppose that $D_k \leq \vartheta^2 \gamma_k^2$, from (28), we have
\[
D_{k+1} \leq (1 - A \beta_k \gamma_k) \gamma_k^2 \vartheta^2 + B \beta_k \gamma_k \vartheta^2 + C \beta_k^2 \leq U_{k+1} \leq \vartheta^2 \gamma_{k+1}^2,
\]
which can be written as
\[
(\chi_k - A)^2 \vartheta^2 + B \vartheta + C \beta_k \gamma_k^3 \leq 0,
\]
(78)
with $\chi_k = \frac{1 - (\frac{\gamma_{k+1}}{\beta_k \gamma_k})^2}{\beta_k \gamma_k} > 0$ as defined in (32). By assumptions presented in Theorem 3, $\chi_k - A < 0$, and we can deduce from the inequality (78) that $\vartheta \geq \vartheta_k$, with
\[
\vartheta \geq \vartheta_k = \frac{B}{2(A - \chi_k)} + \sqrt{\left( \frac{B}{2(A - \chi_k)} \right)^2 + \frac{C \beta_k \gamma_k^3}{A - \chi_k}}.
\]
Recall that both $B$ and $C/B^2\gamma_k^3$ are positive by definition. Consider $\epsilon_1$ and $\epsilon_2$ as defined in (33), we have
\[
2\vartheta_k \leq B \left( \frac{A}{A - \epsilon_1} + \sqrt{\frac{B}{(A - \epsilon_1)^2} + 4C\epsilon_2} \right) + A - \epsilon_1.
\]
We can thus prove that $D_{k+1} \leq \vartheta^2_{k+1}$ with $\vartheta$ defined in (35).

Then we turn to prove (36). Similar to the previous situation, we have $D_{K_0} \leq \vartheta^{2}_{K_0}$ for any $k \geq K_0$, if $D_k \leq \vartheta^{2}_{k-1}$, then
\[
D_{k+1} \leq (1 - A \beta_k \gamma_k) \frac{\beta_k}{\gamma_k} \vartheta^2 + B (\beta_k \gamma_k)^2 \vartheta + C \beta_k^2.
\]
A sufficient condition to ensure $D_{k+1} \leq \vartheta^2_{k+1}$ is that
\[
(1 - A \beta_k \gamma_k) \frac{\beta_k}{\gamma_k} \vartheta^2 + B (\beta_k \gamma_k)^2 \vartheta + C \beta_k^2 \leq \frac{\beta_{k+1}}{\gamma_{k+1}} \vartheta^2.
\]
Thus $\vartheta$ should satisfy
\[
\left( \frac{\beta_k}{\gamma_k} - \frac{\beta_{k+1}}{\gamma_{k+1}} - A \right) \vartheta^2 + B \beta_k^\frac{3}{2} \gamma_k^2 + C \leq 0
\]
If $\frac{\beta_k}{\gamma_k} - \frac{\beta_{k+1}}{\gamma_{k+1}} - A < \beta_k \gamma_k$, then the condition on $\vartheta$ should be $\vartheta \geq \sqrt{\beta_k}$
\[
\vartheta_k = \sqrt{\beta_k - \frac{B \beta_k^\frac{3}{2} \gamma_k^2 + C}{2(\gamma_k - \vartheta_k)}},
\]
where $\vartheta_k = \beta_k^\frac{3}{2} \gamma_k^2 \left( 2 \frac{\beta_k}{\gamma_k} - \frac{\beta_{k+1}}{\gamma_{k+1}} \right) > 0$ as defined in (33). Consider $\epsilon_3$ and $\epsilon_4$ given in (33), we have
\[
\vartheta_k \leq \frac{B \epsilon_4 + \sqrt{(B \epsilon_4)^2 + 4C(A - \epsilon_3)}}{2(A - \epsilon_3)},
\]
then we can prove (37).

\[J. \text{ Proof of Theorem 4}\]

From Theorem 3, we can see that the order of the convergence rate mainly relies on $\nu_1$ and $\nu_2$, as $\gamma_k^3 \propto k^{-2\nu_2}$ and $\beta_k \propto k^{-(\nu_1-\nu_2)}$. However, before the conclusion, we should still verify two points:

1. $i)$ whether the conditions $\epsilon_1 < A$ and $\epsilon_3 < A$ are satisfied;
2. $ii)$ whether the constant terms $\theta$ and $\vartheta$ are bounded.

We start with an elementary and useful result.

**Lemma 6:** For any $a, b, x \in (0, 1)$, we always have
\[
g(x) = x^{-a} \left( 1 - (1 + x)^{-b} \right) < b.
\]
Besides, $\lim_{x \to 0} g(x) = b$ as $a = 1$.

**Proof:** Since $x^{-a} \leq x^{-1}$ for any $x \in (0, 1)$ and $a \in (0, 1]$, we have $g(x) \leq x^{-1} \left( 1 - (1 + x)^{-b} \right) = h(x)$. We calculate the derivative of $h(x)$, i.e.,
\[
h'(x) = x^{-2} \left( \frac{((b+1) x + 1) (1 + x)^{-b-1} - 1}{x} \right).
\]
Thus the monotonicity of $h(x)$ depends on whether $s(x)$ is positive or negative. We further evaluate
\[
s'(x) = -b (b+1) x (1 + x)^{-b-2} \leq 0,
\]
as $b > 0$ and $x > 0$. Thus $s(x)$ is a decreasing function of $x$ over $(0, 1]$. It is easy to get $\lim_{x \to 0} s(x) = 0$, hence, there should be $s(x) < 0$ and $h'(x) < 0$, $\forall x \in (0, 1]$. We have
\[
h(x) = \lim_{x \to 0} h(x) = \frac{1 - (1 + x)^{-b}}{x} = \frac{1}{b},
\]
which concludes the proof.

With the help of Lemma 6, we can easily verify $\epsilon_1 < A$ and $\epsilon_3 < A$.

**Lemma 7:** Consider $\beta_k$ and $\gamma_k$ with forms given in (38), there exists always bounded $\beta_0$ and $\gamma_0$ to guarantee $\epsilon_1 < A$ and $\epsilon_3 < A$.

**Proof:** Consider (38), $\epsilon_1$ can be bounded
\[
\epsilon_1 = \max_{k \geq K_0} \frac{1 - (1 + k^{-4})^{2\nu_2}}{\beta_0 \gamma_0 k^{-\nu_1 - \nu_2}} \leq \max_{x \in (0, 1]} \frac{1 - (1 + x)^{-2\nu_2}}{\beta_0 \gamma_0 x^{\nu_1 + \nu_2}} < 2\nu_2 \beta_0^{-1} \gamma_0^{-1},
\]
where $(a)$ is obtained by the application of Lemma 6 with $a = \nu_1 + \nu_2$ and $b = 2\nu_2$. Therefore, we conclude that $\epsilon_1 < A$ if $\beta_0 \gamma_0 \geq 2\nu_2 / A$. Similarly, we can show that $\epsilon_3 < A$ if $\beta_0 \gamma_0 \geq (\nu_1 - \nu_2) / A$, as
\[
\epsilon_3 = \max_{k \geq K_0} \frac{1 - (1 + k^{-4})^{-(\nu_1 - \nu_2)}}{\beta_0 \gamma_0 k^{-\nu_1 - \nu_2}} < \frac{\nu_1 - \nu_2}{\beta_0 \gamma_0},
\]
The remaining work is to verify whether the constant term in of the convergence rate can be bounded; and our main result stated in Theorem 4 can be easily obtained.

We start with the analysis of $\epsilon_2$ and $\epsilon_4$, i.e.,
\[
\epsilon_2 = \beta_0 \gamma_0^{-3} \max_{k \geq K_0} k^{-\nu_1 - 3\nu_2} \leq \left\{ \begin{array}{ll}
\beta_0 \gamma_0^{-3} K_0^{-\nu_1 - 3\nu_2}, & \text{if } \nu_1 \geq 3\nu_2, \\
\infty, & \text{if } \nu_1 < 3\nu_2,
\end{array} \right.
\]
and
\[
\epsilon_4 = \beta_0^{-1} \gamma_0 \frac{2 \nu_1 - 3\nu_2}{K_0^{\nu_1 - 3\nu_2}} \leq \left\{ \begin{array}{ll}
\beta_0^{-1} \gamma_0 \frac{2 \nu_1 - 3\nu_2}{K_0^{\nu_1 - 3\nu_2}}, & \text{if } \nu_1 \leq 3\nu_2, \\
\infty, & \text{if } \nu_1 > 3\nu_2.
\end{array} \right.
\]
We can see that $\epsilon_2$ and $\epsilon_4$ cannot be bounded under the same condition, unless $\nu_1 = 3\nu_2$.

When $\nu_1 > 3\nu_2$, $\epsilon_2$ is bounded. We can say that $\theta$ is also bounded by its definition, as long as $\beta_0 \gamma_0 \geq 2\nu_2 / A$ (recall Lemma 7). Meanwhile, $\vartheta \to \infty$ as $\epsilon_4 \to \infty$, which makes the bound (36) loose. Therefore, there exists some bounded constant $\Omega_1$, such that $D_k \leq \Omega_1 k^{-2\nu_2}$.

When $\nu_1 < 3\nu_2$, $\epsilon_4$ is bounded whereas $\epsilon_2 \to \infty$. Then there exists $\Omega_2 < \infty$, such that $D_k \leq \Omega_2 k^{-(\nu_1 - \nu_2)}$ if $\beta_0 \gamma_0 \geq (\nu_1 - \nu_2) / A$.

When $\nu_1 = 3\nu_2$, both $\epsilon_2$ and $\epsilon_4$ are bounded. A similar result can be obtained, which concludes the proof.
K. Proof of Theorem 5
In this proof, we consider \( \beta_k = \beta_0 k^{-\frac{3}{4}} \) and \( \gamma_k = \gamma_0 k^{-\frac{1}{4}} \). Our aim is to find the optimal values of \( \beta_0 \) and \( \gamma_0 \) that minimize the constant terms \( \bar{\theta}_2^2 \gamma_0^2 \) and \( \bar{\theta}_2^2 \beta_0 \gamma_0^{-1} \) presented in the upper bounds (34) and (36). According to (35) and (37), we mainly have to minimize
\[
\sqrt{\Omega_1} = \frac{B + \sqrt{B^2 + 4C\epsilon_2 (A - \epsilon_1)}}{2(A - \epsilon_1)} \gamma_0, \\
\sqrt{\Omega_2} = \frac{B\epsilon_4 + \sqrt{(B\epsilon_4)^2 + 4C(A - \epsilon_3)}}{2(A - \epsilon_3)} \beta_0 \gamma_0^{-1}.
\]
From (79) and (80), we find that \( \epsilon_1 = \epsilon_3 < (2\beta_0 \gamma_0)^{-1} \) as \( \nu_1 = \frac{3}{4} \) and \( \nu_2 = \frac{1}{4} \). We also have \( \epsilon_2 = \beta_0 \gamma_0^{-3} \) and \( \epsilon_4 = \beta_0^{-\frac{3}{4}} \gamma_0^{\frac{1}{4}} \). In this special case, it is straightforward to deduce that
\[
\sqrt{\Omega_1} = \sqrt{\Omega_2} = \frac{B\sqrt{\gamma_0} + \sqrt{B^2\gamma_0^2 + 4C\beta_0 (A - \frac{1}{2\sqrt{\gamma_0}})}}{2(A - \frac{1}{2\sqrt{\gamma_0}})}.
\]
(81)
thus the two upper bounds are in fact identical.
We introduce \( x = \frac{1}{\beta_0 \gamma_0} > 0 \) and \( y = \gamma_0 > 0 \) in (81) to lighten the notations, then (81) can be lower bounded as follows:
\[
\Omega_1 = \Omega_2 = \left( \frac{B y + \sqrt{B^2 y^2 + (4C A - 2^{-\frac{1}{4}}x)^2}}{2(A - 2^{-\frac{1}{4}}x)} \right)^2.
\]
(82)
\[
\geq \left( \frac{1}{2(A - 2^{-\frac{1}{4}}x)} \right)^2 \left[ \frac{3}{4} B \gamma_0 A^{-\frac{2}{4}} \left( \frac{A - 2^{-\frac{1}{4}}x}{x} \right) \right]^2
\]
\[
= \left( 2^{-\frac{3}{4}} B \gamma_0 A^{-\frac{2}{4}} \left( \frac{3}{4} A - 2^{-\frac{1}{4}}x \right) \right)^2.
\]
(83)
where \( \frac{y}{x} = \frac{\beta_0}{2A} \) in (a); (b) is by the fact that for any \( a > 0, b > \gamma_0 \), and \( y > 0 \), the fundamental function \( f_1(y) = ay + \sqrt{a^2 y^2 + by^2} \) takes its minimum value when \( y^* = \frac{a^2 b}{a^2 + b} \) and
\[
f_1(y) \geq f_1(y^*) = \left( \frac{3}{4} + \sqrt{\frac{3}{4} + \frac{3}{4}} \right) a^\frac{1}{4} b^\frac{3}{4} = \frac{3}{4} a^\frac{1}{4} b^\frac{3}{4},
\]
here the equality of (b) holds when
\[
y^* = \left( \frac{4C A - 2^{-\frac{1}{4}}x}{3B^2} \right)^{\frac{1}{4}};
\]
(84)
\[
(x^*) = \frac{A}{2}.
\]
(85)
Considering the definition of \( x \) and \( y \), and combining (84) and (85), we obtain
\[
\beta_0 = (x^*)^{-1} = 2^{-\frac{3}{4}} A^{-\frac{1}{4}} B^{-\frac{1}{4}} C^{\frac{3}{4}};
\]
(86)
Then the values of \( A, B, \) and \( C \) defined in (29) or in (22) can be substituted into (83) and (86) to obtain the results presented in Theorem 5.

REFERENCES
[1] W. Li, M. Assaad, and P. Duhamel, “Distributed stochastic optimization in networks with low informational exchange,” in Proc. 55th Annu. Allerton Conf. Commun., Control, Comput. (Allerton), Oct. 2017, pp. 1160–1167.
[2] M. Chiang, P. Hande, T. Lan, and C. W. Tan, “Power control in wireless cellular networks,” Found. Trends Netw., vol. 2, no. 4, pp. 381–533, Apr. 2008.
[3] K.-K. Leung and C. Wan Sung, “An opportunistic power control algorithm for cellular network,” IEEE/ACM Trans. Netw., vol. 14, no. 3, pp. 470–478, Jun. 2006.
[4] N. U. Hassan, M. Assaad, and H. Tembine, “Distributed H∞-based power control in a dynamic wireless network environment,” IEEE Commun. Lett., vol. 17, no. 6, pp. 1124–1127, Jun. 2013.
[5] F. Rashid-Farrokhi, L. Tassiulas, and K. J. R. Liu, “Joint optimal power control and beamforming in wireless networks using antenna arrays,” IEEE Trans. Commun., vol. 46, no. 10, pp. 1313–1324, Oct. 1998.
[6] E. Matskani, N. Sidiropoulos, Z.-Q. Luo, and L. Tassiulas, “Convex approximation techniques for joint multilayer downlink beamforming and admission control,” IEEE Trans. Wireless Commun., vol. 7, no. 7, pp. 2682–2693, Jul. 2008.
[7] C. Wang, H.-M. Wang, D. W. K. Ng, X.-G. Xia, and C. Liu, “Joint beamforming and power allocation for secrecy in peer-to-peer relay networks,” Proc. IEEE Trans. Wireless Commun., vol. 14, no. 6, pp. 3280–3293, Jun. 2015.
[8] Z. Sheng, H. D. Tuan, T. Q. Duong, and H. V. Poor, “Joint power allocation and beamforming for energy-efficient two-way multi-relay communications,” IEEE Trans. Wireless Commun., vol. 16, no. 10, pp. 6660–6671, Oct. 2017.
[9] L. Chen, S. H. Low, and J. C. Doyle, “Random access game and medium access control design,” IEEE/ACM Trans. Netw., vol. 18, no. 4, pp. 1303–1316, Aug. 2010.
[10] B. Liu, P. Terlecky, A. Bar-Noy, R. Govindan, M. J. Neely, and D. Rawitz, “Optimizing information credibility in social swarming applications,” IEEE Trans. Parallel Distrib. Syst., vol. 23, no. 6, pp. 1147–1158, Jun. 2012.
[11] M. J. Neely, “Distributed stochastic optimization via correlated scheduling,” IEEE/ACM Trans. Netw., vol. 24, no. 2, pp. 759–772, Apr. 2016.
[12] K. G. Jamieson, R. Nowak, and B. Recht, “Query complexity of derivative-free optimization,” in Proc. Adv. Neural Inf. Process. Syst., 2012, pp. 2672–2680.
[13] O. Shamir, “On the complexity of bandit and derivative-free stochastic convex optimization,” in Proc. Conf. Learn. Theory, 2013, pp. 3–24.
[14] J. C. Duchi, M. I. Jordan, M. J. Wainwright, and A. Wibisono, “Optimal rates for zero-order convex optimization: The power of two function evaluations,” IEEE Trans. Inf. Theory, vol. 61, no. 5, pp. 2788–2806, May 2015.
[15] A. D. Flaxman, A. T. Kalai, and H. B. McMahan, “Online convex optimization in the bandit setting: Gradient descent without a gradient,” in Proc. 16th Annu. ACM-SIAM Symp. Discrete Algorithms, Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2005, pp. 385–394.
[16] E. Hazan and K. Levy, “Bandit convex optimization: Towards tight bounds,” in Proc. Adv. Neural Inf. Process. Syst., 2014, pp. 784–792.
[17] B. Bauer, Y. T. Lee, and R. Eldan, “Kernel-based methods for bandit convex optimization,” in Proc. 49th Annu. ACM SIGACT Symp. Theory Comput., Jun. 2017, pp. 72–85.
[18] J. Snyman, Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Base Algorithms, vol. 97, Cham, Switzerland: Springer, 2005.
[19] J. Tsitsiklis, D. Bertsekas, and M. Athans, “Distributed asynchronous deterministic and stochastic gradient optimization algorithms,” IEEE Trans. Autom. Control, vol. AC-31, no. 9, pp. 803–812, Sep. 1986.
[20] A. Nedic and D. P. Bertsekas, “Incremental subgradient methods for non-differentiable optimization,” SIAM J. Optim., vol. 12, no. 1, pp. 109–138, Jan. 2001.

[21] D. P. Bertsekas, “Incremental gradient, subgradient, and proximal methods for convex optimization: A survey,” Optim. Mach. Learn., vol. 2010, nos. 1–38, p. 3, 2011.

[22] M. Bennis, S. M. Perlaza, P. Blasco, Z. Han, and H. V. Poor, “Self-organization in small cell networks: A reinforcement learning approach,” IEEE Trans. Wireless Commun., vol. 12, no. 7, pp. 3202–3212, Jul. 2013.

[23] S. Lasaulce and H. Tembine, Game Theory and Learning For Wireless Networks: Fundamentals and Applications. New York, NY, USA: Academic, 2011.

[24] V. S. Borkar, Stochastic Approximation: A Dynamical Systems Viewpoint. Cambridge, U.K.: Cambridge Univ. Press, 2008.

[25] H. J. Kushner and D. S. Clark, Stochastic Approximation Methods for Constrained and Unconstrained Systems, vol. 26. Cham, Switzerland: Springer, 2012.

[26] A. Agarwal and J. C. Duchi, “Distributed delayed stochastic optimization,” in Proc. Adv. Neural Inf. Process. Syst., 2011, pp. 873–881.

[27] N. L. Roux, M. Schmidt, and F. R. Bach, “A stochastic gradient method with an exponential convergence rate for finite training sets,” in Proc. Adv. Neural Inf. Process. Syst., 2012, pp. 2663–2671.

[28] M. Schmidt, N. Le Roux, and F. Bach, “Minimizing finite sums with the stochastic average gradient,” Math. Program., vol. 162, nos. 1–2, pp. 83–112, Mar. 2017.

[29] P. Mertikopoulos, E. V. Belmega, R. Negrel, and L. Sanguineti, “Distributed stochastic optimization via matrix exponential learning,” IEEE Trans. Signal Process., vol. 65, no. 9, pp. 2277–2290, May 2017.

[30] L. M. Rios and N. V. Sahinidis, “Derivative-free optimization: A review of algorithms and comparison of software implementations,” J. Global Optim., vol. 56, no. 3, pp. 1247–1293, Jul. 2013.

[31] J. C. Spall, “Multivariate stochastic approximation using a simultaneous perturbation gradient approximation,” IEEE Trans. Autom. Control, vol. 37, no. 3, pp. 332–341, Mar. 1992.

[32] J. C. Spall, “A one-measurement form of simultaneous perturbation stochastic approximation,” Automatica, vol. 33, no. 1, pp. 109–112, Jan. 1997.

[33] P. Friedrich, M. Krstic, and T. Basar, “Nash equilibrium seeking in noncooperative games,” IEEE Trans. Autom. Control, vol. 57, no. 5, pp. 1192–1207, May 2012.

[34] A. F. Hanif, H. Tembine, M. Assaad, and D. Zeghlache, “On the convergence of a Nash seeking algorithm with stochastic state dependent payoff,” 2012, arXiv:1210.0193. [Online]. Available: http://arxiv.org/abs/1210.0193

[35] C. Manzie and M. Krstic, “Extremum seeking with stochastic perturbations,” IEEE Trans. Autom. Control, vol. 54, no. 3, pp. 580–585, Mar. 2009.

[36] A. Agarwal, O. Dekel, and L. Xiao, “Optimal algorithms for online convex optimization with multi-point bandit feedback,” in Proc. COLT, 2010, pp. 28–40.

[37] S. Liu, B. Kailkhura, P.-Y. Chen, P. Ting, S. Chang, and L. Amini, “Zeroth-order stochastic variance reduction for nonconvex optimization,” in Proc. Adv. Neural Inf. Process. Syst., 2018, pp. 3731–3741.

[38] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, “Robust stochastic approximation approach to stochastic programming,” SIAM J. Optim., vol. 19, no. 4, pp. 1574–1609, Jan. 2009.

[39] P. Mertikopoulos and E. V. Belmega, “Learning to be green: Robust energy efficiency maximization in dynamic MIMO–OFDM systems,” IEEE J. Sel. Areas Commun., vol. 34, no. 4, pp. 743–757, Apr. 2016.

[40] C. W. Tan, M. Chiang, and R. Srikant, “Fast algorithms and performance bounds for sum rate maximization in wireless networks,” IEEE/ACM Trans. Netw., vol. 21, no. 3, pp. 706–719, Jun. 2013.

[41] R. Irmer et al., “Coordinated multipoint: Concepts, performance, and field trial results,” IEEE Commun. Mag., vol. 49, no. 2, pp. 102–111, Feb. 2011.

[42] J. L. Doob, Stochastic Processes. New York, NY, USA: Wiley, 1953.

[43] H. Robbins and S. Monro, “A stochastic approximation method,” Ann. Math. Statist., vol. 22, pp. 400–407, Sep. 1951.