Distributed Stochastic Optimization in Networks with Low Informational Exchange

Wenjie Li, Mohamad Assaad, Pierre Duhamel

Distributed Stochastic Optimization in Networks with Low Informational Exchange

Wenjie Li*†, *Member*, *IEEE*, Mohamad Assaad*†, *Senior Member*, *IEEE*, and Pierre Duhamel*, *Fellow*, *IEEE*
*Laboratoire des Signaux et Systèmes (L2S, UMR CNRS 8506), CentraleSupélec, France
†TCL Chair on 5G, CentraleSupélec, France

***Abstract*—We consider the problem of distributed stochastic optimization in networks. Each node adjusts its action in order to optimize the global utility of the network, which is defined as the sum of local utilities of all nodes. Since the computation of the gradient may require much information exchange, we consider here that each node only has a noisy numerical observation of its local utility. This assumption is quite realistic, especially when the system is too complicated or constantly changing. At each time, nodes may exchange the observation of their numerical local utilities to estimate the global utility. Under the assumptions whether each node has collected the local utilities of all the other nodes or only part of these utilities, we propose two stochastic perturbation based distributed algorithms. We use tools from stochastic approximation to prove that both algorithms converge to the optimum. We then apply our algorithm to a power control problem in wireless networks and present numerical results that corroborate our claim.**

I. Introduction

Distributed optimization is an important problem in networks. The goal is to maximize some predefined objective function for the sake of the performance of the system. Significant amount of work have been done to solve the optimization problems in various applications, for example, power control [1], [2] and beamforming allocation [3], [4].

This paper consider an optimization problem in a distributed network where each node aims to maximize the global utility of the system by adjusting its own action. The global utility is the sum of the local utilities of all nodes of the network. Gradient descent method is the most common technique to deal with optimization problems. In many scenarios in practice, however, the computation of gradient may require too much information exchange between the nodes, an example is provided in Section IV. Furthermore, there are other contexts also where the utility function of each node does not have a closed form expression or the expression is very complex which makes it very hard to use in the optimization, *e.g.*, computation of the derivatives is very complicated or not possible. In this paper, we consider therefore that a node only has a noisy numerical observation of its utility function, which is quite realistic when the system is complex and time-varying. The nodes can only exchange the observation of their local utilities so that each node can have the knowledge of the whole network. However, a node may not receive all the local utilities of the other nodes due to the network topology or other practical issues, *e.g.*, it is not possible to exchange much signaling information. In this situation, a node has to estimate the global utility with only incomplete information of local utilities. We have also taken in account such issue in this paper.

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; 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.

In this paper, we develop distributed algorithms to optimize the global average utility of a network, where the nodes can only exchange the numerical observation of their local utility. The algorithms are closely related to extremum seeking using stochastic perturbation. Two different versions of algorithms are proposed, depending on whether each node has successfully collected the local utilities of all the other nodes or a part of the utilities of other nodes. *To the best of our knowledge, the latter situation has not been considered previously.* We have proved the convergence of the algorithms in both situations, using stochastic approximation tools. Our theoretical results are further justified by simulations.

The rest of the paper is organized as follows. Section II discusses some related work highlights our main contribution. Section III presents the system model and some basic assumptions. Section IV shows a motivating example to explain the interest of our problem. Section V proposes an initial version of our distributed optimization algorithm and proves its convergence. Section VI considers the situation where a node has incomplete information of the global utility of the network. Section VII provides 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 performed using gradient ascent or descent method [5]. This method can achieve a local optimum or global optimum in some special cases (e.g. concavity of the payoff, etc.) of the optimization problem. A distributed asynchronous stochastic gradient optimization algorithms is

presented in [6]. Incremental sub-gradient methods for non-differentiable optimization are discussed in [7]. Interested readers are referred to a survey by Bertsekas [8] 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. This problem is known as derivative-free optimization, see [9] and the references therein. Our goal is then to develop an algorithm that requires only the knowledge of a numerical observation of the utility function. The obtained algorithm must be 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 [10], [11] and the references therein for more details, while we do not consider non-cooperative games in this paper.

Stochastic approximation (SA) [12], [13] is an efficient method to solve the optimization problems in noisy environment. Typically, the action is updated as follows

$$\boldsymbol{a}_{k+1} = \boldsymbol{a}_k + \beta_k \widehat{\boldsymbol{g}}_k. \tag{1}$$

where $\widehat{\boldsymbol{g}}_k$ represents an estimation of the gradient $\boldsymbol{g}_k$. An important assumption is that the estimation error $\boldsymbol{\varepsilon}_k = \widehat{\boldsymbol{g}}_k - \boldsymbol{g}_k$ is seen as a zero-mean random vector with finite variance, for example, see [14]. If the step-size β_k is properly chosen, then $\boldsymbol{a}_k$ can tend to its optimum point asymptotically. The challenge of our work is how to propose such estimation of the gradient only with the noisy numerical observation of the utilities.

Most of the previous work related to derivative-free optimization consider a control center that updates the entire action vector during the algorithm, see [9] 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) [15] can be an option to solve our distributed derivative-free optimization problem. The SPGA algorithm was initially proposed to accelerate the convergence speed of the centralized multi-variate optimization problem with deterministic objective function. 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

$$\widehat{g}_{i,k} = \frac{f\left(\boldsymbol{a}_k + \gamma_k \boldsymbol{\Delta}_k\right) - f\left(\boldsymbol{a}_k - \gamma_k \boldsymbol{\Delta}_k\right)}{2\gamma_k \Delta_{i,k}}, \tag{2}$$

where $\gamma_k > 0$ is vanishing and $\boldsymbol{\Delta}_k = [\Delta_{1,k}, \ldots, \Delta_{N,k}]$ with each element $\Delta_{i,k}$ zero mean and i.i.d. 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 an one-measurement version of the SPGA algorithm in [16] with

$$\widehat{g}_{i,k} = \frac{f\left(\boldsymbol{a}_k + \gamma_k \boldsymbol{\Delta}_k\right)}{2\gamma_k \Delta_{i,k}}. \tag{3}$$

Such algorithm also leads $\boldsymbol{a}_k$ to converge, while 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 γ_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), which may seriously affect the performance of approximation when the value γ_k is too small. As a consequence, the SPGA algorithm is not suitable to solve our stochastic optimization problem.

Extremum seeking (ES) is an interesting approach to deal with non-model based real-time optimization with continuous action spaces. Thanks to the stability results developed in [17], ES is widely used in many applications, see [18], [19] for example. Periodic (sinusoidal) perturbation has been used in most of the ES algorithms, The authors in [20] 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 [21], the authors extended the work in [20] to the case of discrete time and stochastic state-dependent utility functions, convergence to a close region of Nash equilibrium has been proved. However, in a distributed setting, it is challenging to ensure that the sine perturbation of different nodes satisfy 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. Extremum seeking with stochastic perturbation (ESSP) has been proposed in [22] to solve an optimization problem. The ESSP algorithm given by

$$\boldsymbol{a}_{k+1} = \boldsymbol{a}_k + \beta \boldsymbol{v}_k f\left(\boldsymbol{a}_k + \boldsymbol{v}_k\right), \tag{4}$$

with $\boldsymbol{v}_k$ the zero-mean stochastic perturbation. The behavior of the ESSP algorithm has been analyzed in [22], however, under the assumption that the objective function is static and quadratic. Our proposed algorithm is different from (4) as we use the 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 incomplete information of the local utilities of other nodes.

III. System Model

We consider a network composed of a set of nodes denoted by $\mathcal{N}$ with $|\mathcal{N}| = N$. At each time instant k, each node i has a *local* utility function $u_i\left(\boldsymbol{a}_k, \mathbf{S}_k\right)$, which depends on action vector $\boldsymbol{a}_k = [a_{1,k}, \ldots, a_{N,k}]^T \in \mathcal{A}$ of all the nodes and the environment state matrix $\mathbf{S}_k \in \mathcal{S}$. In this work, we consider

$\mathbf{S}_k$ as an i.i.d. ergodic stochastic process and we assume that $\mathcal{S} \subseteq \mathbb{R}^{N\times N}$. As we will see in Section IV, the elements of $\mathbf{S}_k$ may represent the time-varying channel gain between all the pairs of transmitters and receivers.

Assume that each node i only has a noisy numerical estimation of its local utility function, rather than its closed-form expression. Denote $\widetilde{u}_{i,k}$ as the noisy numerical observation of $u_i(\boldsymbol{a}_k, \mathbf{S}_k)$, *i.e.*,

$$\widetilde{u}_{i,k} = u_i(\boldsymbol{a}_k, \mathbf{S}_k) + \eta_{i,k}, \tag{5}$$

where $\eta_{i,k}$ is assumed to be some zero-mean noise with bounded variance.

Introduce $f(\boldsymbol{a}_k, \mathbf{S}_k) = \sum_{i\in\mathcal{N}} u_i(\boldsymbol{a}_k, \mathbf{S}_k)$ as the *global* utility function of the network and $F(\boldsymbol{a}) = \mathbb{E}_{\mathbf{S}}(f(\boldsymbol{a}, \mathbf{S}))$ as the expected value of f. The objective of our work is to design some distributed algorithm to allow each node i to choose an optimal action a_i^* to maximize $F(\boldsymbol{a})$, *i.e.*,

$$\boldsymbol{a}^* = \arg\max_{\boldsymbol{a}} F(\boldsymbol{a}) = \arg\max_{\boldsymbol{a}} \mathbb{E}_{\mathbf{S}}(f(\boldsymbol{a}, \mathbf{S})), \tag{6}$$

only based on the knowledge of $\widetilde{u}_{i,k}$s.

Throughout this paper, we have the following assumptions:

- *A1*. (properties of u_i) $\boldsymbol{a} \longmapsto u_i(\boldsymbol{a}, \mathbf{S})$ is Lipschitz with Lipschitz constant $L_{\mathbf{S}}$, *i.e.*,

$$\|u_i(\boldsymbol{a}, \mathbf{S}) - u_i(\boldsymbol{a}', \mathbf{S})\| \leq L_{\mathbf{S}} \|\boldsymbol{a} - \boldsymbol{a}'\|, \forall i \in \mathcal{N}. \tag{7}$$

  Furthermore, $\mathbf{S} \longmapsto u_i(\boldsymbol{a}, \mathbf{S})$ is integrable with respect to $\mathbf{S}$ so that $F(\boldsymbol{a}) = \mathbb{E}_{\mathbf{S}}\left(\sum_{i\in\mathcal{N}} u_i(\boldsymbol{a}, \mathbf{S})\right) < \infty$.
- *A2*. (properties of F) Both the first and the second order partial derivatives of F exist continuously. There exists $\boldsymbol{a}^*$ such that $\frac{\partial F}{\partial a_i}(\boldsymbol{a}^*) = 0$ and $\frac{\partial^2 F}{\partial a_i^2}(\boldsymbol{a}^*) < 0$, $\forall i \in \mathcal{N}$. Besides,

$$\left|\frac{\partial^2 F}{\partial a_i \partial a_j}(\boldsymbol{a})\right| \leq \alpha_1, \quad \forall i, j \in \mathcal{N}. \tag{8}$$

- *A3*. (properties of η) $\mathbb{E}(\eta_{i,k}) = 0$, $\mathbb{E}\left(\eta_{i,k}^2\right) = \alpha_4$, and $\mathbb{E}(\eta_{i,k}\eta_{j,k}) = 0$ if $i \neq j$.

Remark 1. Assumption *A1* states that $u_i(\boldsymbol{a}, \mathbf{S})$ is a smooth function for any node i. As a consequence, $f(\boldsymbol{a}, \mathbf{S})$ is also Lipschitz with Lipschitz constant $NL_{\mathbf{S}}$

$$\begin{aligned}\|f(\boldsymbol{a}, \mathbf{S}) - f(\boldsymbol{a}', \mathbf{S})\| &= \left\|\sum_{i\in\mathcal{N}} (u_i(\boldsymbol{a}, \mathbf{S}) - u_i(\boldsymbol{a}', \mathbf{S}))\right\| \\ &\leq \sum_{i\in\mathcal{N}} \|u_i(\boldsymbol{a}, \mathbf{S}) - u_i(\boldsymbol{a}', \mathbf{S})\| \leq NL_{\mathbf{S}} \|\boldsymbol{a} - \boldsymbol{a}'\|. \end{aligned} \tag{9}$$

Assumption *A2* means that $\boldsymbol{a}^*$ is a maximizer of $\boldsymbol{a} \longmapsto F(\boldsymbol{a})$ and the objective function $F(\boldsymbol{a})$ is locally concave at $\boldsymbol{a}^*$. In order to lighten the notations, we use

$$F_i'(\boldsymbol{a}) = \frac{\partial F}{\partial a_i}(\boldsymbol{a}), \qquad F_{i,j}''(\boldsymbol{a}) = \frac{\partial^2 F}{\partial a_i \partial a_j}(\boldsymbol{a}),$$

and $\nabla F(\boldsymbol{a}) = [F_i'(\boldsymbol{a}), \ldots, F_N'(\boldsymbol{a})]$ in the rest of the paper.

IV. Motivating Example

We consider a wireless network with N links: each link is a transmitter-receiver pair. Hence a link here corresponds to a node in the system model in Section III. Each transmitter communicates with its corresponding receiver and introduces an interference on the other receivers. Denote $s_{ij,k}$ as the channel gain between transmitter i and receiver j at time k. In this example, the action $a_{i,k}$ is the transmission power of node i at time k. We consider a simple utility function

$$u_i(\boldsymbol{a}_k, \mathbf{S}_k) = \omega \log\left(1 + \log\left(1 + \frac{a_{i,k}s_{ii,k}}{\sigma^2 + \sum_{j\neq i} a_{j,k}s_{ji,k}}\right)\right) - \kappa a_{i,k}. \tag{10}$$

Let

$$\begin{aligned}\texttt{MUI}_{i,k} &= \sigma^2 + \sum_{j\neq i} a_{j,k}s_{ij,k}, \\ \texttt{SINR}_{i,k} &= a_{i,k}s_{ii,k}/\texttt{MUI}_{i,k}, \\ r_{i,k} &= \log(1 + \texttt{SINR}_{i,k}).\end{aligned}$$

Then (10) can be written as

$$u_i(\boldsymbol{a}_k, \mathbf{S}_k) = \omega \log(1 + r_{i,k}) - \kappa a_{i,k}. \tag{11}$$

Notice that we aim to maximize a log-function of the bit rate $r_{i,k}$, which is of type proportional fairness. This type of utility function is used to ensure fairness among the nodes in the network.

In order to use the gradient technique, each node should know

$$\begin{aligned}\frac{\partial}{\partial a_{i,k}} f(\mathbf{S}_k, \boldsymbol{a}_k) &= \frac{\omega}{1 + r_{i,k}} \frac{s_{ii,k}}{\sigma^2 + \sum_{j\neq i} a_{j,k}s_{ji,k}} - \kappa \\ &\quad - \sum_{n\in\mathcal{N}} \frac{\omega}{1 + r_{n,k}} \frac{a_{n,k}s_{nn,k}s_{ni,k}}{\left(\sigma^2 + \sum_{j\neq n} a_{j,k}s_{jn,k}\right)\left(\sigma^2 + \sum_j a_{j,k}s_{jn,k}\right)} \\ &= \omega\left(\frac{\texttt{SINR}_{i,k}}{(1 + r_{i,k})\, a_{i,k}} - \sum_{n\in\mathcal{N}} \frac{\texttt{SINR}_{n,k}}{1 + r_{n,k}} \frac{s_{ni,k}}{\texttt{MUI}_{n,k}(1 + \texttt{SINR}_{n,k})}\right) - \kappa.\end{aligned} \tag{12}$$

We can see that the computation of the exact value of the derivative needs too much exchanges of information: apart from SINR, node i needs to know the values of the channel gain $s_{ni,k}$ $\forall n \in \mathcal{N}$ as well as $a_{n,k}s_{nn,k}$ $\forall n \in \mathcal{N}$ (in order to get $\texttt{MUI}_{n,k}$). Moreover, the channel gain at time k is not possible to estimate as it is constantly changing.

For these reasons, it is important to propose an algorithm in the situation where limited information can be exchanged in the network. For instance, we consider that the nodes can only exchange the numerical values of u_i. In the next two sections, we present our distributed optimization algorithms as well as their performance, in the situations where each node has the *complete* and *incomplete* knowledge of the local utilities of the other nodes, respectively.

V. Distributed optimization algorithm using stochastic perturbation

This section presents an initial version of our distributed optimization algorithm. We consider an unconstrained optimization problem, *i.e.*, $\mathcal{A} = \mathbb{R}^N$.

At the beginning of each iteration, nodes exchange the numerical observation of their local utility function, which is the result of the action performed in the last iteration. In this section, we assume that each node has the *complete* information of $\widetilde{u}_{i,k}$, $\forall i \in \mathcal{N}$, in order to calculate

$$\begin{aligned}\widetilde{f}\left(\boldsymbol{a}, \mathbf{S}_k\right) &= \sum_{i\in\mathcal{N}} \widetilde{u}_{i,k} = \sum_{i\in\mathcal{N}} \left(u_i\left(\boldsymbol{a}, \mathbf{S}_k\right) + \eta_{i,k}\right) \\ &= f\left(\boldsymbol{a}, \mathbf{S}_k\right) + \sum_{i\in\mathcal{N}} \eta_{i,k}. \end{aligned} \tag{13}$$

A. Algorithm

The distributed optimization algorithm using stochastic perturbation (DOSP) is presented in Algorithm 1. For an arbitrary reference node i, the algorithm is given by

$$a_{i,k+1} = a_{i,k} + \beta_k \Phi_{i,k} \widetilde{f}\left(\boldsymbol{a}_k + \gamma_k \boldsymbol{\Phi}_k, \mathbf{S}_k\right), \tag{14}$$

where k denotes the index of iteration, both β_k and γ_k are vanishing, $\Phi_{i,k}$ is randomly generated, and $\boldsymbol{\Phi}_k = [\Phi_{1,k}, \ldots, \Phi_{N,k}]$. Notice that the action performed by each node is the sum of the updated $a_{i,k}$ with a random perturbation $\gamma_k \Phi_{i,k}$. More properties of β_k and $\Phi_{i,k}$ will be discussed later.

Remark 2. Notice that the sine perturbation based extremum seeking method [20], [21] considers a deterministic sine function as the perturbation term. While we use a random perturbation $\boldsymbol{\Phi}_k$ in (14). Comparing (4) and (14), we can see that the amplitude of random perturbation is vanishing in our algorithm, which is not the case in the algorithm presented in [22].

Notice that (14) can be written in the classical form (1) with

$$\widehat{g}_{i,k} = \Phi_{i,k} \widetilde{f}\left(\boldsymbol{a}_k + \gamma_k \boldsymbol{\Phi}_k, \mathbf{S}_k\right) \tag{15}$$

an estimation of the partial derivative $\partial F / \partial a_i$. As stated in Lemma 4, this estimation is asymptotically unbiased if the parameters satisfy the following assumptions:

- *A4*. (properties of $\boldsymbol{\Phi}_k$) The elements of $\boldsymbol{\Phi}_k$ are i.i.d. and mutually independent, each element $\Phi_{i,k}$ is zero-mean and
$$\mathbb{E}\left(\Phi_{i,k}^2\right) = \alpha_2, \quad |\Phi_{i,k}| \leq \alpha_3.$$
- *A5*. (properties of β_k and γ_k) Both β_k and γ_k take real positive values and tend to 0 as k tends to infinity, besides,
$$\sum_{k=1}^{\infty} \beta_k \gamma_k = \infty, \qquad \sum_{k=1}^{\infty} \beta_k^2 < \infty.$$

Algorithm 1 DOSP Algorithm for each node i

1) Initialize $k = 0$ and set the action $a_{i,0}$ randomly.
2) Generate a random variable $\Phi_{i,k}$, perform action $a_{i,k} + \gamma_k \Phi_{i,k}$.
3) Estimate $\widetilde{u}_{i,k}$, exchange its value with the other nodes and calculate $\widetilde{f}\left(\boldsymbol{a}_k + \gamma_k \boldsymbol{\Phi}_k, \mathbf{S}_k\right) = \sum_{j\in\mathcal{N}} \widetilde{u}_{i,k}$.
4) Update $a_{i,k+1}$ according to equation (14).
5) $k = k + 1$, go to 2.

Remark 3. The conditions on β_k and γ_k can be easily achieved. For example, let $\beta_k = k^{-c_1}$ and $\gamma_k = k^{-c_2}$ with the constant $c_1, c_2 \in \mathbb{R}^+$, so that both β_k and γ_k are vanishing. If $c_1 > 0.5$, then $\sum_{k=1}^{\infty} \beta_k^2$ converges. If $c_1 + c_2 < 1$ then $\sum_{k=1}^{\infty} \beta_k \gamma_k$ diverges. Clearly, there exist pairs of c_1 and c_2 to make β_k and γ_k meet the conditions in *A5*.

B. Convergence results

This section investigates the asymptotic behavior of Algorithm 1.

Let $\overline{\boldsymbol{g}}_k$ denote the expected value of $\widehat{\boldsymbol{g}}_k$ with respect to $\boldsymbol{\Phi}$, $\mathbf{S}$, and $\boldsymbol{\eta}$, *i.e.*,

$$\overline{\boldsymbol{g}}_k = \mathbb{E}_{\mathbf{S},\boldsymbol{\Phi},\boldsymbol{\eta}}\left(\widehat{\boldsymbol{g}}_k\right). \tag{16}$$

We rewrite (14) in the generalized Robbins-Monro form [13], *i.e.*,

$$\begin{aligned}\boldsymbol{a}_{k+1} &= \boldsymbol{a}_k + \beta_k \widehat{\boldsymbol{g}}_k \\ &= \boldsymbol{a}_k + \beta_k \left(\alpha_2 \gamma_k \nabla F\left(\boldsymbol{a}_k\right) + \overline{\boldsymbol{g}}_k - \alpha_2 \gamma_k \nabla F\left(\boldsymbol{a}_k\right) + \widehat{\boldsymbol{g}}_k - \overline{\boldsymbol{g}}_k\right) \\ &= \boldsymbol{a}_k + \alpha_2 \beta_k \gamma_k \left(\nabla F\left(\boldsymbol{a}_k\right) + \boldsymbol{b}_k + \frac{\boldsymbol{e}_k}{\alpha_2 \gamma_k}\right), \end{aligned} \tag{17}$$

where

$$\boldsymbol{b}_k = \frac{\overline{\boldsymbol{g}}_k}{\alpha_2 \gamma_k} - \nabla F\left(\boldsymbol{a}_k\right), \tag{18}$$

$$\boldsymbol{e}_k = \widehat{\boldsymbol{g}}_k - \overline{\boldsymbol{g}}_k, \tag{19}$$

represent the estimation bias and the stochastic noise respectively. The following lemmas present some important properties of $\boldsymbol{b}_k$ and $\boldsymbol{e}_k$.

Lemma 4. *If the assumptions A1-A5 hold, then* $\|\boldsymbol{b}_k\| \to 0$ *as* $k \to \infty$.

Proof: See Appendix A. ∎

Lemma 5. *If the assumptions A1-A5 hold and* $\|\boldsymbol{a}_k\| < \infty$ *, then*

$$\lim_{K\to\infty} \mathbb{P}\left(\sup_{K'\geq K} \left\| \sum_{k=K}^{K'} \beta_k \boldsymbol{e}_k \right\| \geq \rho\right) = 0, \quad \forall \rho > 0. \tag{20}$$

Proof: See Appendix B. ∎

Based on the results in Lemma 4 and in Lemma 5, we can build conditions under which $\boldsymbol{a}_k \to \boldsymbol{a}^*$ almost surely.

Theorem 6. *If the assumptions A1-A5 hold and* $\|\boldsymbol{a}_k\| < \infty$ *almost surely, then* $\boldsymbol{a}_k \to \boldsymbol{a}^*$ *as* $k \to \infty$ *almost surely by applying Algorithm 1.*

Proof: We may follow the steps in [15] to complete the proof. From Lemma 5, we can say that the accumulation of the stochastic noise, *i.e.*, $\sum_{k=1}^{\infty} \beta_k \boldsymbol{e}_k$, converges almost surely if all the mentioned conditions are satisfied. Then, from the basic result of stochastic approximation [13], we obtain that $\nabla F\left(\boldsymbol{a}_k\right) + \boldsymbol{b}_k \to \mathbf{0}$ almost surely, as long as the condition on the step size is satisfied, *i.e.*, $\sum_{k=1}^{\infty} \beta_k \gamma_k = \infty$. From Lemma 4, the bias term $\boldsymbol{b}_k$ is vanishing.

Due to the fact that both $\nabla F\left(\boldsymbol{a}_k\right) + \boldsymbol{b}_k$ and $\boldsymbol{b}_k$ tend to zero, we get that $\nabla F\left(\boldsymbol{a}_k\right) \to \mathbf{0}$. Thus $\boldsymbol{a}_k \to \boldsymbol{a}^*$ almost surely. ∎

VI. Optimization Algorithm with incomplete information of utilities of other nodes

A limit of Algorithm 1 is that each node needs to know the local utility of all the other nodes. Such issue is significant as there are many nodes in the network. It is thus important to consider the situation where a node only has access to the local utilities of a subset $\mathcal{I}_{i,k}$ of nodes, with $\mathcal{I}_{i,k} \subseteq \mathcal{N} \setminus \{i\}$. Throughout this section, we have the following assumption:

- *A6*). at any iteration k, an arbitrary node i knows the utility $\widetilde{u}_{j,k}$ of another node j with a constant probability p, *i.e.*, the elements contained in the set $\mathcal{I}_{i,k}$ is random, for any $j \neq i$, we have

$$\mathbb{P}\left(j \in \mathcal{I}_{i,k}\right) = p, \qquad \mathbb{P}\left(j \notin \mathcal{I}_{i,k}\right) = 1 - p. \tag{21}$$

Notice that we do not assume any specified network topology and each node i has a different and *independent* set $\mathcal{I}_{i,k}$.

We propose a modified algorithm and then show its asymptotic performance. The algorithm is described in Algorithm 2. The main difference between Algorithm 1 and Algorithm 2 comes from the approximation of the objective function, *i.e.*,

$$\begin{aligned}&\widetilde{f}_i^{(\mathrm{I})}\left(\boldsymbol{a}, \mathbf{S}_k, \mathcal{I}_{i,k}\right)\\ &= \begin{cases}\widetilde{u}_{i,k} + \frac{N-1}{|\mathcal{I}_{i,k}|}\sum_{j\in\mathcal{I}_{i,k}} \widetilde{u}_{j,k}, & \text{if } |\mathcal{I}_{i,k}| \neq 0,\\ 0, & \text{if } |\mathcal{I}_{i,k}| = 0.\end{cases}\end{aligned} \tag{22}$$

Similar to (14), the algorithm is given by

$$a_{i,k+1} = a_{i,k} + \beta_k \widehat{g}_{i,k}^{(\mathrm{I})} = a_{i,k} + \beta_k \Phi_{i,k} \widetilde{f}_i^{(\mathrm{I})}\left(\boldsymbol{a}, \mathbf{S}_k, \mathcal{I}_{i,k}\right) \tag{23}$$

The basic idea is to consider $(N-1)\sum_{j\in\mathcal{I}_{i,k}} \widetilde{u}_{j,k}/\left|\mathcal{I}_{i,k}\right|$ as a surrogate function of $\sum_{j\in\mathcal{N}\setminus\{i\}} \widetilde{u}_{j,k}$, in the case where the set $\mathcal{I}_{i,k}$ is non-empty. Otherwise, node i does not know any utility of the other nodes, it then cannot estimate the global utility of the system. As a result, node i keeps its previous action, *i.e.*, $a_{i,k+1} = a_{i,k}$.

Algorithm 2 DOSP algorithm for each node i with incomplete information of the utilities of other nodes

1) Initialize $k = 0$ and set the action $a_{i,0}$ randomly.
2) Generate a random variable $\Phi_{i,k}$, perform action $a_{i,k} + \gamma_k \Phi_{i,k}$.
3) Estimate $\widetilde{u}_{i,k}$, exchange its value with the other nodes and calculate $\widetilde{f}_i^{(\mathrm{I})}$ using (22) based on the collected local utilities.
4) Update $a_{i,k+1}$ according to equation (23).
5) $k = k + 1$, go to 2.

The following lemma is useful for our convergence analysis.

Lemma 7. *The expected value of $\widetilde{f}_i^{(\mathrm{I})}\left(\boldsymbol{a}, \mathbf{S}_k, \mathcal{I}_{i,k}\right)$ over all possible sets $\mathcal{I}_{i,k}$ is proportional to $\widetilde{f}\left(\boldsymbol{a}, \mathbf{S}_k\right)$, i.e.,*

$$\mathbb{E}_{\mathcal{I}_{i,k}}\left(\widetilde{f}_i^{(\mathrm{I})}\left(\boldsymbol{a}, \mathbf{S}_k, \mathcal{I}_{i,k}\right)\right) = \left(1 - (1-p)^N\right)\widetilde{f}\left(\boldsymbol{a}, \mathbf{S}_k\right). \tag{24}$$

Proof: See Appendix C. ∎

Introduce $q = 1 - (1-p)^N$ to simplify the notation. Similar to (17), we rewrite (23) as

$$a_{i,k+1} = a_{i,k} + \alpha_2 q \beta_k \gamma_k \left(F_i'\left(\boldsymbol{a}_k\right) + b_{i,k}^{(\mathrm{I})} + \frac{e_{i,k}^{(\mathrm{I})}}{\alpha_2 q \gamma_k}\right) \tag{25}$$

with

$$b_{i,k}^{(\mathrm{I})} = \frac{\mathbb{E}_{\mathbf{S},\boldsymbol{\Phi},\boldsymbol{\eta},\mathcal{I}_{i,k}}\left(\widehat{g}_{i,k}^{(\mathrm{I})}\right)}{\alpha_2 q \gamma_k} - F_i'\left(\boldsymbol{a}_k\right), \tag{26}$$

$$e_{i,k}^{(\mathrm{I})} = \frac{\widehat{g}_{i,k}^{(\mathrm{I})} - \mathbb{E}_{\mathbf{S},\boldsymbol{\Phi},\boldsymbol{\eta},\mathcal{I}_{i,k}}\left(\widehat{\boldsymbol{g}}_k^{(\mathrm{I})}\right)}{\alpha_2 q \gamma_k}. \tag{27}$$

We can follow similar steps as in Section V-B to investigate the convergence of the algorithm.

Theorem 8. *In the situation where nodes do not have the access to all the other nodes' local utilities and the objective function is approximated by applying (22), then we still have $\boldsymbol{a}_k \to \boldsymbol{a}^*$ as $k \to \infty$ almost surely by applying Algorithm 2, as long as the assumptions A1-A5 hold and $\|\boldsymbol{a}_k\| < \infty$ almost surely.*

Proof: See Appendix D. ∎

Remark 9. Although the asymptotic convergence still holds, the convergence speed is reduced if the information of the objective function is incomplete. By comparing (17) and (25), we can see that the equivalent step size is decreased by q times. Besides, the variance of the stochastic noise $e_{i,k}^{(\mathrm{I})}$ is higher, as the randomness is more significant when we use $n < N$ random symbols to represent the average of N symbols.

VII. Simulation Results

In this section, we apply our algorithm to a power control problem as introduced in Section IV in order to have some numerical results. We consider (10) as the local utility function of each node. The time varying channel h_{ij} between node i (transmitter) and node j (receiver) is generated using a Gaussian distribution with variance $\sigma_{ii}^2 = 1$ and $\sigma_{ij}^2 = 1$, $\forall i \neq j$. Notice that the channel gain is $s_{ij} = |h_{ij}|^2$. Besides, we set $\sigma^2 = 1$, $\omega = 10$ and $\kappa = 2$.

In all the simulations (applying different algorithms), we consider $N = 4$ nodes, the step size follows $\beta_k = 0.2k^{-0.3}$, and the initial values of $a_{0,i}$ ($\forall i \in \mathcal{N}$) are generated uniformly in the interval $(0, 20]$. In both Algorithm 1 and Algorithm 2, $\gamma_k = 0.7k^{-0.05}$ and $\Phi_{i,k}$ follows the symmetrical Bernoulli distribution, *i.e.*, $\mathbb{P}\left(\Phi_{i,k} = 1\right) = \mathbb{P}\left(\Phi_{i,k} = -1\right) = 0.5$, $\forall k, i$.

We first compare our proposed algorithm with the sine perturbation based algorithm in [21], considering the situation in which every node has access to all the other nodes' local utilities. The sine perturbation based algorithm has a similar shape as our stochastic perturbation algorithm, with the perturbation term $\Phi_{i,k}$ replaced by a sine function $\lambda_i \sin\left(\Omega_i t_k + \phi_i\right)$ where $t_k = \sum_{k'=1}^{k} \beta_{k'}$, $\Omega_i \neq \Omega_{i'}$ and $\Omega_{i'} + \Omega_i \neq \Omega_{i''}$ $\forall i, i', i''$. In the simulation, we set $\Omega_1 = 4.5$, $\Omega_2 = 5$, $\Omega_3 = 4$, $\Omega_4 = 3.5$, $\lambda_i = 0.4$ and $\phi_i = 0$, $\forall i \in \mathcal{N}$. Notice that this algorithm 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 algorithm using the exact gradient calculated by (12) which is costly to be obtained in practice as discussed in Section IV.

We have performed 100 independent simulations to obtain the average results shown in Figures 1 and 2. Figure 1 represents the utility function $f(\boldsymbol{a}, \mathbf{S})$ as a function of number of iterations. We find that our algorithm converges faster than the reference algorithm proposed in [21] [1]. Figure 2 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 the model with symmetric parameters. We find the oscillation of the power (action) is more significant by applying the reference algorithm.

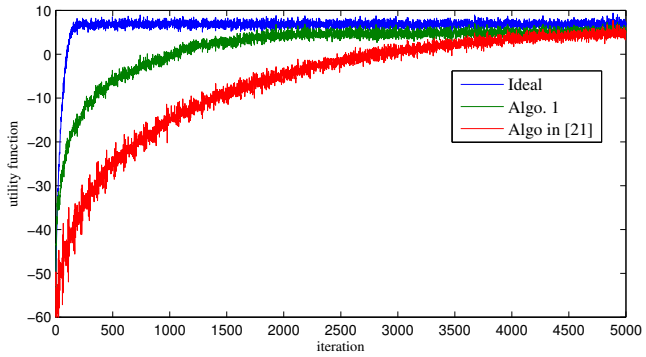

Figure 1. Evolution of the utility function, average results by 100 simulations

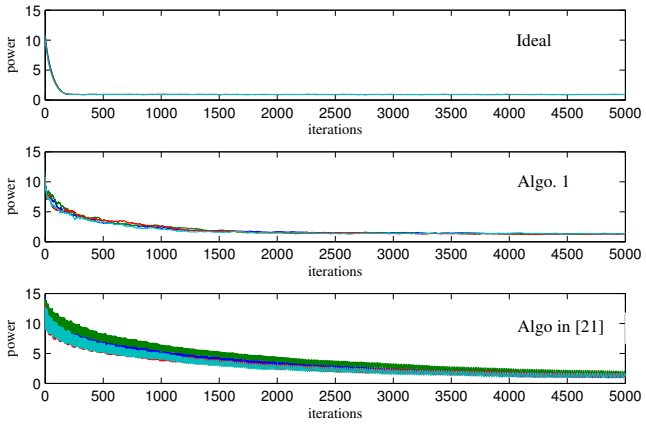

Figure 2. Evolution of power (action) of 4 nodes, average results by 100 simulations

Now we consider the situation where each node has incomplete collection of local utilities. We perform 100 independent simulation with $p \in \{1, 0.5, 0.25, 0.1\}$. Recall that p defined in (21) represents the level of incompleteness. The results are shown in Figure 3. We can see that the convergence speed decreases as the value of p goes smaller. Such influence is not significant even if $p = 0.5$, *i.e.*, a node has only 50% chance to know the local utility of another user, which reduces a lot the information exchange in the network. As the value of p is very small, *i.e.*, $p = 0.1$, same trend of convergence can be observed, although the algorithm converges slowly. On the figure, we show the results obtained after up to 10^4 iterations only, which explains why for $p = 0.1$ the algorithm has not converged yet to the optimal solution.

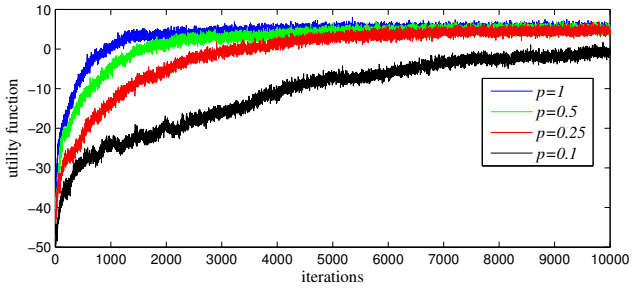

Figure 3. Evolution of the utility function, average results by 100 simulations, with $p \in \{1, 0.5, 0.25, 0.1\}$.

VIII. Conclusion

In this paper we have a challenging distributed optimization problems, under the assumption that only a numerical value of the stochastic state-dependent local utility function of the node is available at each time and nodes need to exchange their local values to optimize the total utilities of the network. We have developed two fully distributed algorithms that converge to the optimum, in the situations where each node has the knowledge of all or only a part of the local utilities of the others. The convergence of our algorithm is examined by studying our algorithm using stochastic approximation technique. Numerical results are also provided for illustration.

Appendix

A. Proof of Lemma 4

By definition, we have

$$\begin{aligned}\overline{\boldsymbol{g}}_k &= \mathbb{E}_{\mathbf{S},\boldsymbol{\Phi},\boldsymbol{\eta}}\left(\boldsymbol{\Phi}_k\left(f\left(\boldsymbol{a}_k+\gamma_k\boldsymbol{\Phi}_k,\mathbf{S}_k\right)+\sum_{i\in\mathcal{N}}\eta_{i,k}\right)\right)\\ &= \mathbb{E}_{\boldsymbol{\Phi}}\left(\boldsymbol{\Phi}_k\mathbb{E}_{\mathbf{S}}\left(f\left(\boldsymbol{a}_k+\gamma_k\boldsymbol{\Phi}_k,\mathbf{S}_k\right)\right)\right)\\ &= \mathbb{E}_{\boldsymbol{\Phi}}\left(\boldsymbol{\Phi}_k F\left(\boldsymbol{a}_k+\gamma_k\boldsymbol{\Phi}_k\right)\right), \end{aligned} \tag{28}$$

recall that the additive noise $\eta_{i,k}$ is zero mean and F is the expected value of f by definition.

Based on Taylor's theorem and mean-valued theorem, there exists $\widetilde{\boldsymbol{a}}_k$ locating between $\boldsymbol{a}_k$ and $\boldsymbol{a}_k + c_k\boldsymbol{\Phi}_k$ such that

$$\begin{aligned}F\left(\boldsymbol{a}_k+\gamma_k\boldsymbol{\Phi}_k\right) = F\left(\boldsymbol{a}_k\right)+\sum_{j\in\mathcal{N}}\gamma_k\Phi_{j,t}F'_j\left(\boldsymbol{a}_k\right)\\ +\sum_{j_1,j_2\in\mathcal{N}}\frac{\gamma_k^2}{2}\Phi_{j_1,k}\Phi_{j_2,k}F''_{j_1,j_2}\left(\widetilde{\boldsymbol{a}}_k\right). \end{aligned} \tag{29}$$

[1] The reference algorithm is quite sensitive to the parameters, the presented results are the best that we have found so far.

Benefit from the properties of $\mathbf{\Phi}_k$ (*A4*), we have, $\forall i \in \mathcal{N}$,

$$\begin{aligned}\overline{g}_{i,k} &= F\left(\boldsymbol{a}_k\right)\mathbb{E}_{\mathbf{\Phi}}\left(\Phi_{i,k}\right) + \gamma_k \sum_{j\in\mathcal{N}} F_j'\left(\boldsymbol{a}_k\right)\mathbb{E}_{\mathbf{\Phi}}\left(\Phi_{i,k}\Phi_{j,k}\right) \\ &\quad + \sum_{j_1,j_2\in\mathcal{N}} \frac{\gamma_k^2}{2}\mathbb{E}_{\mathbf{\Phi}}\left(\Phi_{i,k}\Phi_{j_1,k}\Phi_{j_2,k}F''_{j_1,j_2}\left(\widetilde{\boldsymbol{a}}_k\right)\right) \\ &= \alpha_2\gamma_k\left(F_i'\left(\boldsymbol{a}_k\right) + b_{i,k}\right), \end{aligned} \tag{30}$$

from (28) and (29), with

$$b_{i,k} = \sum_{j_1,j_2\in\mathcal{N}} \frac{\gamma_k}{2\alpha_2}\mathbb{E}_{\mathbf{\Phi}}\left(\Phi_{i,k}\Phi_{j_1,k}\Phi_{j_2,k}F''_{j_1,j_2}\left(\widetilde{\boldsymbol{a}}_k\right)\right).$$

From assumptions *A2* and *A4*, $|b_{i,k}|$ can be upper bounded by

$$|b_{i,k}| \le \sum_{j_1,j_2\in\mathcal{N}} \frac{\gamma_k}{2\alpha_2}\alpha_3^3\alpha_1 = \gamma_k N^2\frac{\alpha_3^3\alpha_1}{2\alpha_2}, \tag{31}$$

then $\|\boldsymbol{b}_k\| \le \gamma_k N^{\frac{5}{2}}\frac{\alpha_3^3\alpha_1}{2\alpha_2} = O\left(\gamma_k\right)$. Here we may conclude that $\|\boldsymbol{b}_k\| \to 0$ as $\gamma_k \to 0$.

B. Proof of Lemma 5

Since $\widehat{\boldsymbol{g}}_k$ and $\widehat{\boldsymbol{g}}_{k'}$ are independent if $k \ne k'$ and

$$\mathbb{E}_{\mathbf{S},\mathbf{\Phi},\boldsymbol{\eta}}\left(\boldsymbol{e}_i\right) = \mathbb{E}_{\mathbf{S},\mathbf{\Phi},\boldsymbol{\eta}}\left(\widehat{\boldsymbol{g}}_k - \mathbb{E}_{\mathbf{S},\mathbf{\Phi},\boldsymbol{\eta}}\left(\widehat{\boldsymbol{g}}_k\right)\right) = \mathbf{0},$$

the sequence $\left\{\sum_{k=K}^{K'}\gamma_k\boldsymbol{e}_k\right\}_{K'\ge K}$ is martingale. By a martingale inequality [23], we have

$$\begin{aligned}&\mathbb{P}\left(\sup_{K'\ge K}\left\|\sum_{k=K}^{K'}\beta_k\boldsymbol{e}_k\right\| \ge \rho\right) \\ &\le \frac{1}{\rho^2}\mathbb{E}_{\mathbf{S},\mathbf{\Phi},\boldsymbol{\eta}}\left(\left\|\sum_{k=K}^{K'}\beta_k\boldsymbol{e}_k\right\|^2\right) \overset{(a)}{=} \frac{1}{\rho^2}\mathbb{E}_{\mathbf{S},\mathbf{\Phi},\boldsymbol{\eta}}\left(\sum_{k=K}^{K'}\left\|\beta_k\boldsymbol{e}_k\right\|^2\right) \\ &\le \frac{1}{\rho^2}\sum_{k=K}^{\infty}\mathbb{E}_{\mathbf{S},\mathbf{\Phi},\boldsymbol{\eta}}\left(\beta_k^2\left\|\widehat{\boldsymbol{g}}_k - \mathbb{E}_{\mathbf{S},\mathbf{\Phi},\boldsymbol{\eta}}\left(\widehat{\boldsymbol{g}}_k\right)\right\|^2\right) \\ &= \frac{1}{\rho^2}\sum_{k=K}^{\infty}\beta_k^2\left(\mathbb{E}_{\mathbf{S},\mathbf{\Phi},\boldsymbol{\eta}}\left(\left\|\widehat{\boldsymbol{g}}_k\right\|^2\right) - \left\|\mathbb{E}_{\mathbf{S},\mathbf{\Phi},\boldsymbol{\eta}}\left(\widehat{\boldsymbol{g}}_k\right)\right\|^2\right) \\ &\le \frac{1}{\rho^2}\sum_{k=K}^{\infty}\beta_k^2\mathbb{E}_{\mathbf{S},\mathbf{\Phi},\boldsymbol{\eta}}\left(\left\|\widehat{\boldsymbol{g}}_k\right\|^2\right) \end{aligned} \tag{32}$$

where (a) holds as $\mathbb{E}\left(\boldsymbol{e}_{k_1}^T\cdot\boldsymbol{e}_{k_2}\right) = 0$ for any $k_1 \ne k_2$.

We then need to show that the average value of $\|\widehat{\boldsymbol{g}}_k\|^2$ is bounded. For any $i \in \mathcal{N}$, we evaluate

$$\begin{aligned}&\mathbb{E}_{\mathbf{S},\mathbf{\Phi},\boldsymbol{\eta}}\left(\widehat{g}_{i,k}^2\right) \\ &= \mathbb{E}_{\mathbf{S},\mathbf{\Phi},\boldsymbol{\eta}}\left(\left(\Phi_{i,k}f\left(\boldsymbol{a}_k + \gamma_k\mathbf{\Phi}_k, \mathbf{S}_k\right) + \Phi_{i,k}\sum_{j\in\mathcal{N}}\eta_{j,k}\right)^2\right) \\ &\overset{(a)}{=} \mathbb{E}_{\mathbf{S},\mathbf{\Phi}}\left(\left(\Phi_{i,k}f\left(\boldsymbol{a}_k + \gamma_k\mathbf{\Phi}_k, \mathbf{S}_k\right)\right)^2\right) + N\alpha_2\alpha_4 \\ &\overset{(b)}{\le} \alpha_3^2\mathbb{E}_{\mathbf{S},\mathbf{\Phi}}\left(\left(f\left(\boldsymbol{a}_k + \gamma_k\mathbf{\Phi}_k, \mathbf{S}_k\right)\right)^2\right) + N\alpha_2\alpha_4 \\ &\overset{(c)}{\le} \alpha_3^2\mathbb{E}_{\mathbf{S},\mathbf{\Phi}}\left(\left(\left\|f\left(\mathbf{0}, \mathbf{S}_k\right)\right\| + NL_{\mathbf{S}_k}\left\|\boldsymbol{a}_k + \gamma_k\mathbf{\Phi}_k\right\|\right)^2\right) + N\alpha_2\alpha_4 \\ &\overset{(d)}{\le} 2\alpha_3^2\mathbb{E}_{\mathbf{S}}\left(\mu_{\mathbf{S}_k}^2 + N^2L_{\mathbf{S}_k}^2\left(\left\|\boldsymbol{a}_k\right\| + \gamma_k N^{\frac{1}{2}}\alpha_3\right)^2\right) + N\alpha_2\alpha_4 \\ &\overset{(e)}{=} 2\alpha_3^2\left(\mu + N^2L\left(\left\|\boldsymbol{a}_k\right\| + \gamma_k N^{\frac{1}{2}}\alpha_3\right)^2\right) + N\alpha_2\alpha_4 \\ &< \infty \end{aligned} \tag{33}$$

where (a) is due to $\mathbb{E}_{\mathbf{\Phi},\boldsymbol{\eta}}\left(\left(\Phi_{i,k}\sum_{j\in\mathcal{N}}\eta_{j,k}\right)^2\right) = N\alpha_2\alpha_4$ and (b) is by Assumption *A4*. From (9), we have

$$\left\|f\left(\boldsymbol{a}, \mathbf{S}\right)\right\| - \left\|f\left(\mathbf{0}, \mathbf{S}\right)\right\| \le \left\|f\left(\boldsymbol{a}, \mathbf{S}\right) - f\left(\mathbf{0}, \mathbf{S}\right)\right\| \le NL_{\mathbf{S}}\left\|\boldsymbol{a}\right\|,$$

so $\left\|f\left(\boldsymbol{a}, \mathbf{S}\right)\right\| \le \left\|f\left(\mathbf{0}, \mathbf{S}\right)\right\| + L_{\mathbf{S}}\left\|\boldsymbol{a}\right\|$, (c) can be obtained. We denote $\mu_{\mathbf{S}_k} = \left\|f\left(\mathbf{0}, \mathbf{S}_k\right)\right\|$ in (d) and the inequality is because of $(x+y)/2 \le \sqrt{\left(x^2+y^2\right)/2}$, $\forall x, y \in \mathbb{R}$. In (e), we introduce $\mu = \mathbb{E}_{\mathbf{S}}\left(\mu_{\mathbf{S}_k}^2\right)$ and $L = \mathbb{E}_{\mathbf{S}}\left(L_{\mathbf{S}_k}^2\right)$.

Combine (32) and (33), Lemma 5 can be proved as $\lim_{K\to\infty}\sum_{k=K}^{\infty}\beta_k^2 = 0$ by Assumption *A5*.

C. Proof of Lemma 7

We start with the conditional expectation, based on (22), we have

$$\begin{aligned}&\mathbb{E}_{\mathcal{I}_{i,k}}\left(\widetilde{f}_i^{(\mathrm{I})}\left(\boldsymbol{a}, \mathbf{S}_k, \mathcal{I}_{i,k}\right) \middle| \left|\mathcal{I}_{i,k}\right| = n\right) \\ &= \begin{cases}\widetilde{u}_{i,k} + \frac{N-1}{n}\mathbb{E}_{\mathcal{I}_{i,k}}\left(\sum_{j\in\mathcal{I}_{i,k}}\widetilde{u}_{j,k}\middle|\left|\mathcal{I}_{i,k}\right| = n\right), & \text{if } n \ne 0, \\ 0, & \text{if } n = 0. \end{cases}\end{aligned} \tag{34}$$

Denote $\mathcal{U}_i^{(n)}$ as a collection of all possible sets $\mathcal{I}_{i,k}$ such that $\left|\mathcal{I}_{i,k}\right| = n$, *e.g.*, $\mathcal{U}_i^{(1)} = \left\{\{1\}, \ldots, \{i-1\}, \{i+1\}, \ldots\right\}$. Since each node has an equal probability to be involved in $\mathcal{I}_{i,k}$, the sets in $\mathcal{U}_i^{(n)}$ are also equiprobable, *i.e.*,

$$\mathbb{P}\left(\mathcal{I}_{i,k} = \mathcal{I}\middle|\left|\mathcal{I}_{i,k}\right| = n\right) = \frac{1}{\binom{N-1}{n}}, \quad \forall\mathcal{I} \in \mathcal{U}_i^{(n)},$$

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

$$\begin{aligned}&\mathbb{E}_{\mathcal{I}_{i,k}}\left(\sum_{j\in\mathcal{I}_{i,k}}\widetilde{u}_{j,k}\middle|\left|\mathcal{I}_{i,k}\right| = n\right) \\ &= \sum_{\mathcal{I}\in\mathcal{U}_i^{(n)}}\frac{1}{\binom{N-1}{n}}\sum_{j\in\mathcal{I}}\widetilde{u}_{j,k} = \frac{1}{\binom{N-1}{n}}\frac{n\binom{N-1}{n}}{N-1}\sum_{j\in\mathcal{N}\setminus\{i\}}\widetilde{u}_{j,k} \\ &= \frac{n}{N-1}\sum_{j\in\mathcal{N}\setminus\{i\}}\widetilde{u}_{j,k}. \end{aligned} \tag{35}$$

Combine (34) and (35), for any $n \in \{1, \ldots, N-1\}$, we have

$$\mathbb{E}_{\mathcal{I}_{i,k}}\left(\widetilde{f}_i^{(\mathrm{I})}\left(\boldsymbol{a}, \mathbf{S}_k, \mathcal{I}_{i,k}\right) \middle| \left|\mathcal{I}_{i,k}\right| = n\right) = \widetilde{f}\left(\boldsymbol{a}, \mathbf{S}_k\right). \tag{36}$$

According to the basic rule of expectation,

$$\begin{aligned}&\mathbb{E}\left(\widetilde{f}_i^{(\mathrm{I})}\left(\boldsymbol{a}, \mathbf{S}_k, \mathcal{I}_{i,k}\right)\right) \\ &= \sum_{n=0}^{N-1}\mathbb{P}\left(\left|\mathcal{I}_{i,k}\right| = n\right)\mathbb{E}_{\mathcal{I}_{i,k}}\left(\widetilde{f}_i^{(\mathrm{I})}\left(\boldsymbol{a}, \mathbf{S}_k, \mathcal{I}_{i,k}\right) \middle| \left|\mathcal{I}_{i,k}\right| = n\right) \\ &= \sum_{n=1}^{N-1}\mathbb{P}\left(\left|\mathcal{I}_{i,k}\right| = n\right)\widetilde{f}\left(\boldsymbol{a}, \mathbf{S}_k\right) \\ &= \left(1 - \mathbb{P}\left(\left|\mathcal{I}_{i,k}\right| = 0\right)\right)\widetilde{f}\left(\boldsymbol{a}, \mathbf{S}_k\right) \\ &= \left(1 - (1-p)^N\right)\widetilde{f}\left(\boldsymbol{a}, \mathbf{S}_k\right). \end{aligned}$$

Lemma 7 is then proved.

D. Proof of Theorem 8

As the analysis in Section V-B. We mainly need to check: i) whether $b_{i,k}^{(\mathrm{I})}$ is vanishing; ii) whether $\sum_{k=1}^{\infty}\beta_k \boldsymbol{e}_k^{(\mathrm{I})}$ converges. The proof of these two statements are more complicated as compared to Appendix A and Appendix B, since there is an additional random term $\mathcal{I}_{i,k}$ in both $b_{i,k}^{(\mathrm{I})}$ and $e_{i,k}^{(\mathrm{I})}$, compared with $b_{i,k}$ and $e_{i,k}$ discussed in Section V

We start with the bias term, by Lemma 7,

$$\begin{aligned}\mathbb{E}_{\mathbf{S},\boldsymbol{\Phi},\boldsymbol{\eta},\mathcal{I}_{i,k}}\left(\widehat{g}_{i,k}^{(\mathrm{I})}\right) &= \mathbb{E}_{\mathbf{S},\boldsymbol{\Phi},\boldsymbol{\eta},\mathcal{I}_{i,k}}\left(\Phi_{i,k}\widetilde{f}_i^{(\mathrm{I})}\left(\boldsymbol{a},\mathbf{S}_k,\mathcal{I}_{i,k}\right)\right)\\ &= q\mathbb{E}_{\mathbf{S},\boldsymbol{\Phi},\boldsymbol{\eta}}\left(\Phi_{i,k}\widetilde{f}\left(\boldsymbol{a},\mathbf{S}_k\right)\right).\end{aligned}$$

We then find that $b_{i,k}^{(\mathrm{I})}$ in this section is the same as $b_{i,k}$ defined in (18), as

$$b_{i,k}^{(\mathrm{I})} = \frac{q\mathbb{E}_{\mathbf{S},\boldsymbol{\Phi},\boldsymbol{\eta}}\left(\Phi_{i,k}\widetilde{f}\left(\boldsymbol{a},\mathbf{S}_k\right)\right)}{\alpha_2 q\gamma_k} - F_i'\left(\boldsymbol{a}_k\right) = b_{i,k}.$$

Therefore, $b_{i,k}^{(\mathrm{I})}$ can be vanishing as γ_k is vanishing according to Lemma 4.

Then we turn to analyze the stochastic noise. From the definition (27) , the sequence $\left\{\sum_{k=K}^{K'}\gamma_k \boldsymbol{e}_k^{(\mathrm{I})}\right\}_{K'\geq K}$ is martingale. Similar to (32)

$$\begin{aligned}&\mathbb{P}\left(\sup_{K'\geq K}\left\|\sum_{k=K}^{K'}\beta_k \boldsymbol{e}_k^{(\mathrm{I})}\right\|\geq\rho\right)\\ &\leq \frac{1}{\rho^2}\sum_{k=K}^{\infty}\beta_k^2\sum_{i\in\mathcal{N}}\mathbb{E}_{\mathbf{S},\boldsymbol{\Phi},\boldsymbol{\eta},\mathcal{I}_{i,k}}\left(\left(\widehat{g}_{i,k}^{(\mathrm{I})}\right)^2\right).\end{aligned}\tag{37}$$

In order to evaluate the average of $\left(\widehat{g}_{i,k}^{(\mathrm{I})}\right)^2$, we first consider, for any $n\in\{1,\ldots,N-1\}$,

$$\begin{aligned}&\mathbb{E}_{\mathcal{I}_{i,k}}\left(\left(\widehat{g}_{i,k}^{(\mathrm{I})}\right)^2\left|\,|\mathcal{I}_{i,k}|=n\right.\right)\\ &=\Phi_{i,k}^2\mathbb{E}_{\mathcal{I}_{i,k}}\left(\left(\widetilde{u}_{i,k}+\frac{N-1}{n}\sum_{j\in\mathcal{I}_{i,k}}\widetilde{u}_{j,k}\right)^2\middle||\mathcal{I}_{i,k}|=n\right)\\ &\overset{(a)}{\leq}\alpha_3^2\left(n+1\right)\left(\widetilde{u}_{i,k}^2+\left(\frac{N-1}{n}\right)^2\mathbb{E}_{\mathcal{I}_{i,k}}\left(\sum_{j\in\mathcal{I}_{i,k}}\widetilde{u}_{j,k}^2\middle||\mathcal{I}_{i,k}|=n\right)\right)\\ &\overset{(b)}{=}\alpha_3^2\left(n+1\right)\left(\widetilde{u}_{i,k}^2+\frac{N-1}{n}\sum_{j\in\mathcal{N}\setminus\{i\}}\widetilde{u}_{j,k}^2\right)\\ &\overset{(c)}{\leq}\alpha_3^2\left(N\widetilde{u}_{i,k}^2+2\left(N-1\right)\sum_{j\in\mathcal{N}\setminus\{i\}}\widetilde{u}_{j,k}^2\right),\end{aligned}\tag{38}$$

where (a) is by $\Phi_{i,k}^2\leq\alpha_3^2$ and $\sum_{i=1}^m x_i/m\leq\sqrt{\sum_{i=1}^m x_i^2/m}$, (b) can be proved using (35), and (c) is due to $(n+1)/n\leq 2$ as $1\leq n\leq N-1$.

Notice that the upper bound in (38) does not depend on n and that $\widehat{g}_{i,k}^{(\mathrm{I})}=0$ as $n=0$, hence

$$\mathbb{E}_{\mathcal{I}_{i,k}}\left(\left(\widehat{g}_{i,k}^{(\mathrm{I})}\right)^2\right)\leq\alpha_3^2\left(N\widetilde{u}_{i,k}^2+2\left(N-1\right)\sum_{j\in\mathcal{N}\setminus\{i\}}\widetilde{u}_{j,k}^2\right).$$

Our last target is then to verify $\mathbb{E}_{\mathbf{S},\boldsymbol{\Phi},\boldsymbol{\eta}}\left(\widetilde{u}_{i,k}^2\right)$ is bounded for any $i\in\mathcal{N}$, which can be proved using similar steps as in (33).

REFERENCES

[1] K.-K. Leung and C. W. Sung, "An opportunistic power control algorithm for cellular network," *IEEE/ACM Trans. on Networking*, vol. 14, no. 3, pp. 470–478, 2006.

[2] N. U. Hassan, M. Assaad, and H. Tembine, "Distributed h-infinity based power control in a dynamic wireless network environment," *IEEE Communications Letters*, vol. 17, no. 6, pp. 1124–1127, 2013.

[3] F. Rashid-Farrokhi, L. Tassiulas, and K. R. Liu, "Joint optimal power control and beamforming in wireless networks using antenna arrays," *IEEE Trans. on Communications*, vol. 46, no. 10, pp. 1313–1324, 1998.

[4] E. Matskani, N. D. Sidiropoulos, Z.-q. Luo, and L. Tassiulas, "Convex approximation techniques for joint multiuser downlink beamforming and admission control," *IEEE Trans. on Wireless Communications*, vol. 7, no. 7, 2008.

[5] J. Snyman, *Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms*. Springer Science & Business Media, 2005, vol. 97.

[6] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Trans. on Automatic Control*, vol. 31, no. 9, pp. 803–812, 1986.

[7] A. Nedic and D. P. Bertsekas, "Incremental subgradient methods for nondifferentiable optimization," *SIAM Journal on Optimization*, vol. 12, no. 1, pp. 109–138, 2001.

[8] D. P. Bertsekas, "Incremental gradient, subgradient, and proximal methods for convex optimization: A survey," *Optimization for Machine Learning*, vol. 2010, no. 1-38, p. 3, 2011.

[9] L. M. Rios and N. V. Sahinidis, "Derivative-free optimization: a review of algorithms and comparison of software implementations," *Journal of Global Optimization*, vol. 56, no. 3, pp. 1247–1293, 2013.

[10] 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. on Wireless Communications*, vol. 12, no. 7, pp. 3202–3212, 2013.

[11] S. Lasaulce and H. Tembine, *Game theory and learning for wireless networks: fundamentals and applications*. Academic Press, 2011.

[12] V. S. Borkar, "Stochastic approximation: A dynamical systems viewpoint," 2008.

[13] H. J. Kushner and D. S. Clark, *Stochastic Approximation Methods for Constrained and Unconstrained Systems*. Springer Science & Business Media, 2012, vol. 26.

[14] P. Mertikopoulos, E. V. Belmega, R. Negrel, and L. Sanguinetti, "Distributed stochastic optimization via matrix exponential learning," *IEEE Trans. on Signal Processing*, vol. 65, no. 9, pp. 2277–2290, 2017.

[15] J. C. Spall, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation," *IEEE Trans. on Automatic Control*, vol. 37, no. 3, pp. 332–341, 1992.

[16] ——, "A one-measurement form of simultaneous perturbation stochastic approximation," *Automatica*, vol. 33, no. 1, pp. 109–112, 1997.

[17] H.-H. Wang and M. Krstic, "Extremum seeking for limit cycle minimization," *IEEE Trans. on Automatic Control*, vol. 45, no. 12, pp. 2432–2436, 2000.

[18] M. S. Stanković and D. M. Stipanović, "Extremum seeking under stochastic noise and applications to mobile sensors," *Automatica*, vol. 46, no. 8, pp. 1243–1251, 2010.

[19] I. Vandermeulen, M. Guay, and P. J. McLellan, "Discrete-time distributed extremum-seeking control over networks with unstable dynamics," *IEEE Trans. on Control of Network Systems*, 2017.

[20] P. Frihauf, M. Krstic, and T. Basar, "Nash equilibrium seeking in noncooperative games," *IEEE Trans. on Automatic Control*, vol. 57, no. 5, pp. 1192–1207, 2012.

[21] A. Hanif, H. Tembine, M. Assaad, and D. Zeghlache, "On the convergence of a nash seeking algorithm with stochastic state dependent payoff," *arXiv preprint arXiv:1210.0193*, 2012.

[22] C. Manzie and M. Krstic, "Extremum seeking with stochastic perturbations," *IEEE Trans. on Automatic Control*, vol. 54, no. 3, pp. 580–585, 2009.

[23] J. L. Doob, *Stochastic Processes*. Wiley New York, 1953.