IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 15 December 2023, accepted 23 December 2023, date of publication 26 December 2023,
date of current version 4 January 2024.

Digital Object Identifier 10.1109/ACCESS.2023.3347649

RESEARCH ARTICLE

Distributed Newton Step Projection Algorithm for Online Convex Optimization Over Time-Varying Unbalanced Networks

JIAYI WU AND YU-PING TIAN
School of Automation, Hangzhou Dianzi University, Hangzhou, Zhejiang 310018, China
Corresponding author: Yu-Ping Tian (yptian@hdu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 62073107, and in part by the Natural Science Foundation of Zhejiang Province under Grant LZ21F030002.

ABSTRACT In this paper, we investigate distributed online optimization over multi-agent networks, where a group of agents aim to cooperatively seek the minimum value of a time-varying global loss function that can be expressed as the sum of the local loss functions privately owned by a single agent on the network at each iteration. In addition, all agents do not know the future loss function, and information about the loss function is disclosed only after the agent has made a decision. We are interested in scenarios where the communication topology of a multi-agent network is a sequence of time-varying unbalanced graphs and the loss function of each agent is a class of non-strongly convex functions. We generalize the distributed online Newton step algorithm to a more general multi-agent network by introducing a consensual-based auxiliary estimation to rescale the contribution of each agent in optimization. Our algorithm only uses row stochastic matrices and does not require the agent to know the out-degree of its in-neighbors. The convergence of regret bound over time-varying unbalanced networks is rigorously proved. Simulation results also verify the effectiveness of the proposed algorithm, and show its advantage in convergence speed compared with the first-order algorithms.

INDEX TERMS Distributed optimization, online convex optimization, Newton step projection algorithm, multi-agent network, time-varying unbalanced directed graphs.

I. INTRODUCTION

Online convex optimization (OCO) problems has been encountered in many practical applications such as auctions and portfolio management [1], [2], [3]. In these problems time-varying (and possibly adversarial) loss functions are revealed just after the decision has been made. Therefore, the idea of optimizing the cost function is not applicable to such problems, and is replaced by optimizing the so-called regret, which measure the difference between the cost generated by the optimization algorithm and the best fixed decision cost in hindsight [4].

In recent years, distributed OCO has attracted more and more attention due to the fact that many distributed optimization problems emerged in multi-agent systems involve temporal variations in the cost structure and constraints [5]. When the dynamics or the probability distribution of the variations are known *a priori*, distributed dynamic optimization [6] or stochastic optimization methods [7] may be applicable for improving the robustness of the optimization algorithms. However, if the variations in the problem are compiled by an arbitrarily varying cost function revealed over time, and the agent can use the information of the loss function only after the decision-making, the solution to these problems call for effective distributed online optimization algorithms.

The online gradient descent (OGD) method has been proved effective for centralized OCO, see [2], [3], and [8]. Recently, Yan et al. [9] proposed a decentralized online optimization algorithm based on the subgradient method in which the agents interact over a weighted strongly connected directed graph, and the regret bounds of $\mathcal{O}(\sqrt{T})$ and $\mathcal{O}(\log T)$

The associate editor coordinating the review of this manuscript and approving it for publication was Deepak Mishra.

are obtained for convex and strongly convex loss functions, respectively. Hosseini et al. proposed a consensus based distributed online dual averaging algorithm in [5], and later extended the algorithm to time-varying networks in [10]. Inspired by the saddle point dynamics in [11], Mateos-Nunez and Cortes introduced a distributed OCO algorithm over jointly-connected and weight-balanced networks in [12]. Moreover, [13] studied the case when the loss function is strongly pseudoconvex. And [14] investigated the distributed online aggregation optimization problem, that is, the loss function value depends on both the individual decision value and the aggregate information of all individual decision variables.

All the above references considered undirected graphs or balanced digraphs as the communication topology and use double stochastic weight matrix to reach the optimal consensus among agents, which is difficult to apply in many practical cases. Because agents are usually limited by their own power, they have different communication ranges and have different interference and noise patterns. In addition, when packet loss [15] or time-varying communication delay [16] occurs in the network, the dynamic switching of network topology will also occur. An example is the estimation using sensor networks, where the observations of each sensor varies over time due to packet loss. Therefore, it is very meaningful to extend the algorithm to more realistic settings on the unbalanced graph. At present, there are some literatures to solve the distributed OCO problem in view of the unbalance of the network. Akbari et al. [17] combined push-sum protocol [18] and the online subgradient method for unbalanced jointly strongly connected digraphs. But their algorithm uses the column stochastic weight matrix in cooperation, and requires each agent to accurately know its in-neighbors' out-degree. This requirement is difficult to meet in many cases, especially when the agents use a broadcast-based communication scheme. Moreover, this method can not solve the constrained optimization problem with projection. Reference [19] also combined push-sum and primal-dual algorithm to solve the distributed online optimization problem constrained by local set constraints and coupled inequalities while dealing with the unbalance of the network. Compared with the methods mentioned in the above literature, introduce row stochastic matrix is more convenient in practical application. Reference [20] firstly used an auxiliary variable to estimate the left eigenvector of the weight matrix, and the unbalance of the network is overcome by scaling the gradient. Then [21] extended the algorithm to time-varying networks. These methods are also being extended to online Settings. Reference [22] investigates a distributed online constrained optimization problem with differential privacy. In [23], Tada et al. discuss a primal-dual distributed algorithm for online convex optimization with a time-varying coupled constraint. Reference [24] designed a distributed bandit online leaning method by extending the celebrated mirror descent algorithm for a bandit setup. These algorithms are all designed in fixed topology. Different from the above literature, [25] designed online distributed bandit algorithms by exploiting dual averaging method, which can be applied to time-varying unbalanced networks. In addition, there are other literatures that solve the distributed online optimization problem in unbalanced networks. For example, [26] uses the method based on surplus [27]. Reference [28] studies a distributed online constrained optimization problem over time-varying digraphs without explicit subgradients, which uses a sequence of row stochastic and column stochastic weight matrices simultaneously.

The operation of distributed OCO algorithms based on first-order gradient information is computationally simple, but their convergence rate is slow in most cases. It is well known that for the off-line convex optimization problems, Newton's method can achieve faster convergence rate by using the second order gradient information of the cost function. However, computing and storing the Hessian matrix along with its inverse at each iteration will seriously affect the efficiency of online algorithms. To overcome such inconvenience, distributed approximate or quasi-Newton methods were proposed [29], [30], the main idea of which is to generate some estimation matrix such as curvature estimation using the gradient information to replace the inverse of Hessian matrix. Further, a reinforced consensus-based Newton control method is proposed in [31], which integrats the consensus strategy and the latest knowledge across the network into local descent direction, and allows different control levels according to demand. In [32], the Newton step algorithm was extended for the distributed OCO problem to achieve the logarithm convergence of the regret. But it is restricted to the case of time-invariant and undirected networks. In practical applications, due to packet loss and communication delay, communication networks in multi-agent systems are often described by time-varying and unbalanced directed graphs.

Motivated by the above literature analysis, we aim to extend the distributed online Newton step algorithm to the setting of time-varying unbalanced networks. On the one hand, in contrast of [32], which only uses double stochastic matrices and is only applicable to fixed balanced digraphs, we combine the method for time-varying unbalanced digraph in [21], [22], and [25] to rescale the optimization part through consensual-based auxiliary estimation, overcoming the constraint that the network must be weighted and balanced. In addition, compared with [17], we only use a row-stochastic matrix and do not require the agent to know the out-degree of its in-neighbors. And we can solve the constrained online optimization problem by projection method. Our improved version greatly expands the application range of the original algorithm. On the other hand, unlike most first-order algorithms adopted in [9], [10], and [12], only the regret bound of $\sqrt{T}$ can be reached for non-strongly convex functions. Our algorithm does not need to compute and store the Hessian matrix along with its inverse matrix

of the cost function, it only uses the gradient information to achieve the logarithm convergence rate for so-called exp-concave functions which can not be strongly convex. This result weakens the requirement for strongly convex functions in the above paper. In addition, the convergence speed of our algorithm has been greatly improved.

The rest of this paper is organized as follows. In Section II, the preparatory knowledge related to function and graph theory is given, as well as the introduction to distributed OCO problems. In Section III, we propose a distributed online Newton step projection algorithm and analyze the convergence of the regret bound. In section IV, the performance of the algorithm is verified by numerical simulations. Concluding remarks are given in Section V.

II. PRELIMINARIES

A. NOTATION

All vectors are regarded as column vectors. The notation $()^T$ indicates the transpose of a vector or a matrix. 1_n represents the n-dimensional column vector with all elements being one. $\|\cdot\|$ denotes the standard Euclidean norm for a vector, or the spectral norm for a matrix. The symbol e_i represents the vector with i th entry being one and all the others being zero, i.e., $e_i = [0, \cdots, 1_i, \cdots, 0]^T$. For convex function $f(x)$, denote its gradient at x as $\nabla f(x)$. The H-induced projection of x onto $\mathcal{X}$ is defined as

$$\Pi_{\mathcal{X}}^{H}(x) = \arg\min_{y\in\mathcal{X}}(x-y)^T H(x-y), \tag{1}$$

where H is a definitely positive matrix.

B. DISTRIBUTED ONLINE CONVEX OPTIMIZATION PROBLEM

The problem formulation of the centralized OCO with detailed explanations can be found in [2]. Now, let us directly consider the scenario of distributed online optimization. Suppose there is a set of n agents which can communicate with each other on a time-varying network. Denote by $V = \{1, \cdots, n\}$ the set of agent. At each discrete time $t \in \{1, \cdots, T\}$, each agent i makes a decision $x_i(t) \in \mathcal{X} \subseteq \mathbb{R}^m$, where $\mathcal{X}$ is a bounded decision set, based on its previous decision, the loss function revealed in the previous iteration, and the information received from its in-neighbors. After making the decision $x_i(t)$, a locally convex loss function $f_{t,i} : \mathcal{X} \to \mathbb{R}$ will be presented, and this loss function is only known to the agent i. In such a distributed scenario, at each moment t, the goal of the entire network of the agents is to minimize the sum of the loss functions,

$$\min F_t(x) = \sum_{i=1}^{n} f_{t,i}(x), \quad \text{s.t. } x \in \mathcal{X}. \tag{2}$$

Since the loss function is previously unknown to each agent, the goal of the online algorithm is to ensure that the time average of the difference between the loss incurred by the algorithm and the loss of the best fixed decision in hindsight

$$x^* \in \arg\min_{x\in\mathcal{X}} \sum_{t=1}^{T}\sum_{i=1}^{n} f_{t,i}(x) \tag{3}$$

is small. This difference, namely

$$R_T = \sum_{t=1}^{T}\sum_{i=1}^{n}(f_{t,i}(x_i(t)) - f_{t,i}(x^*)), \tag{4}$$

is referred to as the regret of the distributed online optimization algorithm. An online algorithm performs well if its regret is sub-linear as a function of T, i.e., $\lim_{T\to\infty} R_T/T = 0$, because it implies that the time average performance of this algorithm is as well as the best fixed strategy in hindsight.

The following assumptions are made for the problem (2).

Assumption 1: The set $\mathcal{X} \subseteq \mathbb{R}^m$ is non-empty, convex, and compact, and the Euclidean diameter of $\mathcal{X}$, diam $(\mathcal{X}) = \sup_{x,x'\in\mathcal{X}} \|x - x'\|$, is bounded by a constant D.

Assumption 2: For all $i \in V$, function $f_{t,i}(x)$ is differentiable and L-Lipschitz continuous on $\mathcal{X}$, i.e., $\|\nabla f_{t,i}(x)\| \leq L, \forall x \in \mathcal{X}$.

Assumption 3: For all $i \in V$ and all $t \geq 0$, function $f_{t,i} : \mathbb{R}^m \mapsto \mathbb{R}$ is α-exp concave on $\mathcal{X}$, namely, the function $g : \mathcal{X} \mapsto \mathbb{R}$ defined by

$$g(x) = \mathrm{e}^{-\alpha f(x)}$$

is concave.

Note that the α-exp concave function defines a class of functions larger than strongly convex functions, i.e., any strongly convex function is also an α-exp concave function on a bounded domain, but the inverse proposition does not hold true.

Lemma 1 [3]: Let $f : \mathcal{X} \mapsto \mathbb{R}$ be a α-exp concave function, D and L be the diameter and the Lipschitz constant of $\mathcal{X}$, respectively. Then, for all $\beta \leq \frac{1}{2}\min\{\frac{1}{4LD}, \alpha\}$ and all $x, y \in \mathcal{X}$, the following inequality

$$f(x) \geq f(y) + \nabla f(y)^T(x-y) + \frac{\beta}{2}(x-y)^T\nabla f(y)\nabla f(y)^T(x-y)$$

holds.

Remark 1: This lemma shows that for α-exp concave functions, instead of using the Taylor series, we can use a stronger approximation to approximate the cost functions up to the second order, where we do not use the Hessian matrix of the cost function, but only the gradient.

C. COMMUNICATION NETWORK

To solve the distributed OCO problem considered in this paper, each agent must send some necessary information to its out-neighbors. The communication topology among the agents can be described by a time-varying weighted digraph $G(t) = (V, E(t), A(t))$ with the vertex set $V = \{1, \cdots, n\}$ and the edge set $E(t) \subseteq V \times V$. If agent j can send messages to agent i, then $(j, i) \in E(t)$, and node j is referred to as an in-neighbor of node i. Denote by $N_{i,t}^{in} = \{j|(j, i) \in E(t)\}$ the set of all the in-neighbors of node i at time t, and by

$d_{i,t}^{in} = |N_{i,t}^{in}|$ the in-degree node i at time t. Similarly, denote by $N_{i,t}^{out} = \{j|(i,j) \in E(t)\}$ and $d_{i,t}^{out} = |N_{i,t}^{out}|$ the set of out-neighbors and the out-degree, respectively, of node i at time t. If $d_{i,t}^{in} = d_{i,t}^{out}$ holds for all nodes in V, then $G(t)$ is called as a balanced digraph; otherwise, it is called an unbalanced digraph. If there exists a directed path between any pair of nodes in the digraph $G(t)$, then $G(t)$ is said to be strongly connected. The weighted adjacent matrix of $G(t)$ is denoted as $A(t) = [a_{ij}(t)] \in \mathbb{R}^{n\times n}$. If $(j,i) \in E(t)$, then the associated weight $a_{ij}(t)>0$; otherwise, $a_{ij}(t) = 0$. In addition, $A(t)$ is called a row-stochastic matrix if $\sum_{j=1}^{n} a_{ij}(t) = 1$.

The following assumption is made for the communication topology.

Assumption 4: The time-varying communication topology described by $G(t)$ is $\mathbb{B}$-jointly strongly connected, namely, there exists an positive inter $B \in \mathbb{Z}_{>0}$ such that for all $t \in \mathbb{Z}_{>0}$, the joint digraph $G(t)\cup G(t+1)\cdots\cup G(t+B-1)$ is strongly connected.

III. MAIN RESULTS

In this section a distributed Newton step algorithm algorithm is presented to solve the constrained online convex optimization problem (2). Then, the convergence results for the algorithm are rigorously summarized and proved.

A. DISTRIBUTED ONLINE NEWTON STEP PROJECTION ALGORITHM

The problem studied in this paper is that agents are expected to reach a common optimal estimate of the sum of their loss functions, but the equilibrium points of each local loss function are often different. In most existing methods, the adjacent matrix is often required to be double stochastic. In this case, the sum of rows of the adjacent matrix is equal to one provides the possibility that the agents will be able to reach a consensus, and the sum of the columns of the adjacent matrix is equal to one guarantees that each agent's contribution to the network is the same. If the communication topology is unbalanced, the iteration of the agent's decisions needs to be compensated in order to equalize the weight of the contribution made by each agent. Based on this motivation, we present the distributed online Newton step algorithm in Algorithm 1. In the algorithm, each agent $i \in V$ maintains two variables $x_i(t) \in \mathcal{X}$ and $z_i(t) \in \mathbb{R}^m$, where $t \in \{1,\cdots,T\}$ is the discrete time iteration. The first is the decision variation, the second compensates for unbalancedness. In each iteration, each node estimates its Hessian matrix using the gradient of the loss function at the previous time, exchanges information with its neighbors and moves in the direction of the product of the inverse of the estimated Hessian matrix and the gradient vector, and then obtains the decision value through an additional projection operation.

In Algorithm 1, the positive definite symmetric matrix $H_i(t)$ is constructed using only the gradient information, and the matrix $H_i(t)^{-1}$ needs to be calculated for each iteration. The matrix inversion lemma in [33] shows that for the invertible matrix A and the vector x,

$$(A + xx^T)^{-1} = A^{-1} - \frac{A^{-1}xx^TA^{-1}}{1 + x^TA^{-1}x}$$

Thus, given $H_i(t-1)$ and $\nabla_{t,i}$, and $H_i(0) = \epsilon I$ and $H_i(0)^{-1} = \frac{1}{\epsilon}I$, one can simply compute $H_i(t)^{-1}$ using only matrix-vector and vector-vector products. At the same time, because of $H_i(t) = \sum_{r=1}^{t} \nabla_{r,i}\nabla_{r,i}^T + \epsilon I = H_i(t-1) + \nabla_{t,i}\nabla_{t,i}^T$, we do not have to store all the gradient information in the process of iteration, only need to use $H_i(t-1)$ and the gradient $\nabla_{t,i}$ of the current moment to calculate, reducing the storage cost during the algorithm execution.

Algorithm 1 Distributed Online Newton Step Projection Algorithm

Input: convex set $\mathcal{X}$, total iteration number T, parameters $\beta = \frac{1}{2}\min\{\frac{1}{4LD}, \alpha\}$, $\epsilon = \frac{1}{\beta^2D^2}$, initial states $x_i(1) \in \mathcal{X}$, $z_i(0) = e_i \in \mathbb{R}^m$, $H_i(0) = \epsilon I$, $\forall i \in V$.

for $t = 1,\cdots,T$ **do**
  For decision $x_i(t)$ the adversary reveals the loss function $f_{t,i}(x_i(t))$, $\forall i \in V$
  Compute gradient $\nabla_{t,i}$ of function $f_{t,i}(\cdot)$ at $\sum_{j=1}^{n} a_{ij}(t)x_j(t)$, for all $i \in V$
  Update $H_i(t) = H_i(t-1) + \nabla_{t,i}\nabla_{t,i}^T$, $\forall i \in V$
  for all $i \in V$ **do**
    $\hat{x}_i(t+1) = \sum_{j=1}^{n} a_{ij}(t)x_j(t) - \frac{1}{\beta}\frac{H_i(t)^{-1}\nabla_{t,i}}{z_{ii}(t)}$
    $z_i(t+1) = \sum_{j=1}^{n} a_{ij}(t)z_j(t)$
    $x_i(t+1) = \Pi_{\mathcal{X}}^{H_i(t)}(\hat{x}_i(t+1))$
  end for
end for

The projection function used in Algorithm 1 is the H-induced projection defined by (1). The well known Pythagoream Theorem holds for this kind of projection, and is restated as follows.

Lemma 2 [34]: Let $\mathcal{X} \subseteq \mathbb{R}^m$ be a non-empty convex set, under the projection operation $\Pi_{\mathcal{X}}^{H}(x)$ for any $y \in \mathcal{X}$, the following inequality

$$\|x - y\| \geq \|\Pi_{\mathcal{X}}^{H}(x) - y\|$$

holds.

The the weights $a_{ij}(t)$ in Algorithm 1 satisfy the following assumption.

Assumption 5: There exists a positive constant γ such that the weighted adjacent matrix $A(t) = [a_{ij}(t)]$ of the communication digraph $G(t)$ is row-stochastic and satisfies, for all $t \geq 0$ and all $i \in V$, $a_{ij}(t) > \gamma$ if $j \in N_{i,t}^{in}$, otherwise, $a_{ij}(t) = 0$, and $a_{ii}(t) = 1 - \sum_{j\in N_{i,t}^{in}} a_{ij}(t) > \gamma$.

Remark 2: The positive lower bound γ in the above assumption excludes the case that $a_{ij}(t) \to 0$ when $T \to \infty$ if agent i has access to agent j.

B. CONVERGENCE OF THE ALGORITHM

Lemma 3 [19]: Suppose Assumptions 4 and 5 hold. Consider Algorithm 1. For $s \geq k \geq 0$, denote

$A(s:k) = A(s-1)\cdots A(k)$ and let $a_{ij}(s:k)$ be entries of $A(s:k)$. Define $A(k:k) = I$. Then, for all $t \geq 0$, there is a normalized vector $\pi(t)((\pi(t))^T 1_n = 1)$ such that the following statements are true. a)

1) For any $i, j \in V$ and $s \geq k$, inequality $|a_{ij}(s:k) - \pi_j(k)| \leq C\lambda^{s-k}$ holds for $C>0$, $0<\lambda<1$.
2) For any $i \in V$ and $t \geq 0$, relation $\pi_i(t) \geq \eta$ holds for the constant $\eta \geq \gamma(n-1)B$.
3) $(\pi(k))^T = (\pi(k+1))^T A(k)$.
4) $|z_{ii}(k) - \pi_i(0)| \leq C\lambda^k$.

Remark 3: Lemma 3 gives the basic properties of $A(t)$, which provides a sufficient condition for the convergence analysis later. We assume that $A(t)$ is row-stochastic, so it is a valid weight matrix for consensus (see, e.g., [35]). However, different from previous literature, $A(t)$ is not column stochastic, so it is necessary to set another auxiliary variable related to the left Perron eigenvector of the weight matrix.

Theorem 1: Suppose Assumptions 1 - 5 hold. Let $\bar{x}(t) = \sum_{i=1}^n \pi_i(t)x_i(t)$. Then, under Algorithm 1, there exist positive constants D_1, D_2 and D_3 such that

$$\begin{aligned}&\sum_{t=1}^{T}(F_t(\bar{x}(t)) - F_t(x^*))\\ &\leq \frac{LD_1}{D_2}\sum_{t=1}^{T}\sum_{i=1}^{n}\|x_i(t)-\bar{x}(t)\| \\ &\quad + \frac{D_3}{2\beta D_2}\sum_{t=1}^{T}\sum_{i=1}^{n}\nabla_{t,i}^T H_i(t)^{-1}\nabla_{t,i} + \frac{\beta n\epsilon D^2}{2D_2}. \end{aligned} \tag{5}$$

Proof: Denote $y_i(t) = \sum_{j=1}^n a_{ij}(t)x_j(t)$. By line 6 of Algorithm 1 it holds that

$$\begin{aligned}&(\hat{x}_i(t+1)-x^*)^T H_i(t)(\hat{x}_i(t+1)-x^*)\\ &= \left(y_i(t) - \frac{1}{\beta}\frac{H_i(t)^{-1}\nabla_{t,i}}{z_{ii}(t)} - x^*\right)^T H_i(t)\\ &\quad\cdot\left(y_i(t) - \frac{1}{\beta}\frac{H_i(t)^{-1}\nabla_{t,i}}{z_{ii}(t)} - x^*\right)\\ &= (y_i(t)-x^*)^T H_i(t)(y_i(t)-x^*) - \frac{2}{\beta}(y_i(t)-x^*)^T\frac{\nabla_{t,i}}{z_{ii}(t)}\\ &\quad + \frac{1}{\beta^2}\frac{\nabla_{t,i}^T H_i(t)^{-1}\nabla_{t,i}}{z_{ii}^2(t)}. \end{aligned} \tag{6}$$

Since $x_i(t+1)$ is the projection of $\hat{x}_i(t+1)$ on $\mathcal{X}$ under the $H_i(t)$-induced norm, by Lemma 2 we have

$$\begin{aligned}&(x_i(t+1)-x^*)^T H_i(t)(x_i(t+1)-x^*)\\ &\quad\leq (\hat{x}_i(t+1)-x^*)^T H_i(t)(\hat{x}_i(t+1)-x^*). \end{aligned} \tag{7}$$

According to the α-exp concave property of $f_{t,i}(x)$ given by Lemma 1, the parameters given in Algorithm 1 ensure that

$$\begin{aligned}\nabla_{t,i}^T(y_i(t)-x^*) &\geq f_{t,i}(y_i(t)) - f_{t,i}(x^*)\\ &\quad\times\frac{\beta}{2}(y_i(t)-x^*)^T\nabla_{t,i}\nabla_{t,i}^T(y_i(t)-x^*). \end{aligned} \tag{8}$$

Further, since $f_{t,i}$ is L-Lipschitz continuous on $\mathcal{X}$, we have

$$\begin{aligned}|f_{t,i}(y_i(t)) - f_{t,i}(\bar{x}(t))| &\leq L\|y_i(t)-\bar{x}(t)\|\\ &= L\|\sum_{j=1}^{n}a_{ij}(t)(x_j(t)-\bar{x}(t))\|\\ &\leq L\sum_{j=1}^{n}a_{ij}(t)\|x_j(t)-\bar{x}(t)\|. \end{aligned} \tag{9}$$

Using (8) and (9) we can get the upper bound of the second term in the right side of (6) as

$$\begin{aligned}&-\frac{2}{\beta}(y_i(t)-x^*)^T\frac{\nabla_{t,i}}{z_{ii}(t)}\\ &\leq -\frac{2}{\beta z_{ii}(t)}(f_{t,i}(y_i(t)) - f_{t,i}(x^*)\\ &\quad + \frac{\beta}{2}(y_i(t)-x^*)^T\nabla_{t,i}\nabla_{t,i}^T(y_i(t)-x^*))\\ &= -\frac{2}{\beta z_{ii}(t)}(f_{t,i}(y_i(t)) - f_{t,i}(\bar{x}(t)) + f_{t,i}(\bar{x}(t)) - f_{t,i}(x^*))\\ &\quad - \frac{1}{z_{ii}(t)}(y_i(t)-x^*)^T\nabla_{t,i}\nabla_{t,i}^T(y_i(t)-x^*)\\ &\leq \frac{2}{\beta z_{ii}(t)}(|f_{t,i}(y_i(t)) - f_{t,i}(\bar{x}(t))| - (f_{t,i}(\bar{x}(t)) - f_{t,i}(x^*)))\\ &\quad - \frac{1}{z_{ii}(t)}(y_i(t)-x^*)^T\nabla_{t,i}\nabla_{t,i}^T(y_i(t)-x^*)\\ &\leq \frac{2}{\beta z_{ii}(t)}(L\sum_{j=1}^{n}a_{ij}(t)\|x_j(t)-\bar{x}(t)\| - (f_{t,i}(\bar{x}(t)) - f_{t,i}(x^*)))\\ &\quad - \frac{1}{z_{ii}(t)}(y_i(t)-x^*)^T\nabla_{t,i}\nabla_{t,i}^T(y_i(t)-x^*). \end{aligned} \tag{10}$$

Noticing that $z_{ii}(t) \in (0,1)$, we have

$$\begin{aligned}&-\frac{1}{z_{ii}(t)}(y_i(t)-x^*)^T\nabla_{t,i}\nabla_{t,i}^T(y_i(t)-x^*)\\ &\quad\leq -(y_i(t)-x^*)^T\nabla_{t,i}\nabla_{t,i}^T(y_i(t)-x^*). \end{aligned} \tag{11}$$

Considering $H_i(t) = H_i(t-1) + \nabla_{t,i}\nabla_{t,i}^T$, we can combine the right side of the above equation and the first term in (6) and get

$$\begin{aligned}&(y_i(t)-x^*)^T H_i(t)(y_i(t)-x^*)\\ &\quad - (y_i(t)-x^*)\nabla_{t,i}\nabla_{t,i}^T(y_i(t)-x^*)\\ &= (y_i(t)-x^*)^T(H_i(t) - \nabla_{t,i}\nabla_{t,i}^T)(y_i(t)-x^*)\\ &= (y_i(t)-x^*)^T H_i(t-1)(y_i(t)-x^*). \end{aligned} \tag{12}$$

Since the $H_i(t)$-induced norm $\|\cdot\|_{H_i(t)}$ is a convex function because of the definite positiveness of matrix $H_i(t)$, and $\sum_{j=1}^n a_{ij} = 1$, it holds that

$$\begin{aligned}&(y_i(t)-x^*)^T H_i(t-1)(y_i(t)-x^*)\\ &= \|(y_i(t)-x^*)\|_{H_i(t-1)}^2\\ &= \|\sum_{j=1}^{n}a_{ij}(t)(x_j(t)-x^*)\|_{H_i(t-1)}^2 \end{aligned}$$

$$\leq \sum_{j=1}^{n} a_{ij}(t)\|x_j(t)-x^*\|^2_{H_i(t-1)}. \tag{13}$$

Substituting (10) and (13) into (6) and considering (7) yield

$$\begin{aligned}
&(x_i(t+1)-x^*)^T H_i(t)(x_i(t+1)-x^*)\\
&\leq \sum_{j=1}^{n} a_{ij}(t)(x_j(t)-x^*)^T H_i(t-1)(x_j(t)-x^*)\\
&\quad+\frac{2L}{\beta z_{ii}(t)}\sum_{j=1}^{n} a_{ij}(t)\|x_j(t)-\bar{x}(t)\|\\
&\quad-\frac{2}{\beta z_{ii}(t)}(f_{t,i}(\bar{x}(t))-f_{t,i}(x^*))\\
&\quad+\frac{1}{\beta^2 z_{ii}^2(t)}\nabla_{t,i}^T H_i(t)^{-1}\nabla_{t,i}.
\end{aligned} \tag{14}$$

Multiplying both sides of the above equation by $\pi_i(t+1)$ and then summing them for all $i \in V$, we get

$$\begin{aligned}
&\sum_{i=1}^{n}\pi_i(t+1)(x_i(t+1)-x^*)^T H_i(t)(x_i(t+1)-x^*)\\
&\leq \sum_{i=1}^{n}\pi_i(t+1)\sum_{j=1}^{n} a_{ij}(t)(x_j(t)-x^*)^T H_i(t-1)(x_j(t)-x^*)\\
&\quad+\frac{2L}{\beta}\sum_{i=1}^{n}\frac{\pi_i(t+1)}{z_{ii}(t)}\sum_{j=1}^{n} a_{ij}(t)\|x_j(t)-\bar{x}(t)\|\\
&\quad-\frac{2}{\beta}\sum_{i=1}^{n}\frac{\pi_i(t+1)}{z_{ii}(t)}(f_{t,i}(\bar{x}(t))-f_{t,i}(x^*))\\
&\quad+\frac{1}{\beta^2}\sum_{i=1}^{n}\frac{\pi_i(t+1)}{z_{ii}^2(t)}\nabla_{t,i}^T H_i(t)^{-1}\nabla_{t,i}.
\end{aligned} \tag{15}$$

Since $(\pi(t))^T = (\pi(t+1))^T A(t)$ by Lemma 3, the first term of the right side of (15) can be rewritten as

$$\begin{aligned}
&\sum_{i=1}^{n}\pi_i(t+1)\sum_{j=1}^{n} a_{ij}(t)(x_j(t)-x^*)^T H_i(t-1)(x_j(t)-x^*)\\
&=\sum_{i=1}^{n}\pi_i(t)(x_i(t)-x^*)^T H_i(t-1)(x_i(t)-x^*).
\end{aligned} \tag{16}$$

Let

$$D_1 = \sup_{t\geq 0}\sum_{i=1}^{n}\frac{\pi_i(t+1)}{z_{ii}(t)}, \tag{17}$$

$$D_2 = \min_{i\in V}\inf_{t\geq 0}\frac{\pi_i(t+1)}{z_{ii}(t)}, \tag{18}$$

$$D_3 = \max_{i\in V}\sup_{t\geq 0}\frac{\pi_i(t+1)}{z_{ii}^2(t)}. \tag{19}$$

Since $\pi_i(t+1) \geq \eta \geq \gamma(n-1)B$ for any $i \in V$ by Lemma 3, D_1, D_2 and D_3 are all positive constants. Then, we have

$$\begin{aligned}
&\frac{2L}{\beta}\sum_{i=1}^{n}\frac{\pi_i(t+1)}{z_{ii}(t)}\sum_{j=1}^{n} a_{ij}(t)\|x_j(t)-\bar{x}(t)\|\\
&\leq \frac{2LD_1}{\beta}\sum_{i=1}^{n}\|x_i(t)-\bar{x}(t)\|.
\end{aligned} \tag{20}$$

$$\begin{aligned}
&-\frac{2}{\beta}\sum_{i=1}^{n}\frac{\pi_i(t+1)}{z_{ii}(t)}(f_{t,i}(\bar{x}(t))-f_{t,i}(x^*))\\
&\leq -\frac{2D_2}{\beta}(F_t(\bar{x}(t)-F_t(x^*)).
\end{aligned} \tag{21}$$

$$\begin{aligned}
&\frac{1}{\beta^2}\sum_{i=1}^{n}\frac{\pi_i(t+1)}{z_{ii}^2(t)}\nabla_{t,i}^T H_i(t)^{-1}\nabla_{t,i}\\
&\leq \frac{D_3}{\beta^2}\sum_{i=1}^{n}\nabla_{t,i}^T H_i(t)^{-1}\nabla_{t,i}.
\end{aligned} \tag{22}$$

Substituting (16),(20),(21) and (22) into (15) yields

$$\begin{aligned}
&\sum_{i=1}^{n}\pi_i(t+1)(x_i(t+1)-x^*)^T H_i(t)(x_i(t+1)-x^*)\\
&\leq \sum_{i=1}^{n}\pi_i(t)(x_i(t)-x^*)^T H_i(t-1)(x_i(t)-x^*)\\
&\quad+\frac{2LD_1}{\beta}\sum_{i=1}^{n}\|x_i(t)-\bar{x}(t)\|-\frac{2D_2}{\beta}(F_t(\bar{x}(t)-F_t(x^*))\\
&\quad+\frac{D_3}{\beta^2}\sum_{i=1}^{n}\nabla_{t,i}^T H_i(t)^{-1}\nabla_{t,i}
\end{aligned} \tag{23}$$

Moving the terms in the above equation, dividing both sides by $\frac{2D_2}{\beta}$, and then summing on $t = 1, \cdots, T$, we get

$$\begin{aligned}
&\sum_{t=1}^{T}(F_t(\bar{x}(t))-F_t(x^*))\\
&\leq \frac{\beta}{2D_2}\sum_{t=1}^{T}(\sum_{i=1}^{n}\pi_i(t)(x_i(t)-x^*)^T H_i(t-1)(x_i(t)-x^*)\\
&\quad-\sum_{i=1}^{n}\pi_i(t+1)(x_i(t+1)-x^*)^T H_i(t)(x_i(t+1)-x^*))\\
&\quad+\frac{LD_1}{D_2}\sum_{t=1}^{T}\sum_{i=1}^{n}\|x_i(t)-\bar{x}(t)\|\\
&\quad+\frac{D_3}{2\beta D_2}\sum_{t=1}^{T}\sum_{i=1}^{n}\nabla_{t,i}^T H_i(t)^{-1}\nabla_{t,i}.
\end{aligned} \tag{24}$$

Since $H_i(0) = \epsilon I_m$, and $(x_i(T+1)-x^*)^T H_i(T)(x_i(T+1)-x^*) \geq 0$ because of the definite positiveness of $H_i(T)$, we have

$$\begin{aligned}
&\sum_{t=1}^{T}(\sum_{i=1}^{n}\pi_i(t)(x_i(t)-x^*)^T H_i(t-1)(x_i(t)-x^*)\\
&\quad-\sum_{i=1}^{n}\pi_i(t+1)(x_i(t+1)-x^*)^T H_i(t)(x_i(t+1)-x^*))\\
&=\sum_{i=1}^{n}\Big(\pi_i(1)(x_i(1)-x^*)^T H_i(0)(x_i(1)-x^*)\\
&\quad-\pi_i(T+1)(x_i(T+1)-x^*)^T H_i(T)(x_i(T+1)-x^*)\Big)
\end{aligned}$$

$$\leq \sum_{i=1}^{n} \pi_i(1)\epsilon \|x_i(1) - x^*\|^2$$
$$\leq \sum_{i=1}^{n} \pi_i(1)\epsilon D^2 \leq n\epsilon D^2. \quad (25)$$

Substituting (25) into (24) yields

$$\sum_{t=1}^{T}(F_t(\bar{x}(t)) - F_t(x^*)) \leq \frac{\beta n\epsilon D^2}{2D_2} + \frac{LD_1}{D_2}\sum_{t=1}^{T}\sum_{i=1}^{n}\|x_i(t) - \bar{x}(t)\| + \frac{D_3}{2\beta D_2}\sum_{t=1}^{T}\sum_{i=1}^{n}\nabla_{t,i}^T H_i(t)^{-1}\nabla_{t,i}. \quad (26)$$

The proof is completed.

Remark 4: Theorem 1 establishes the optimization part of convergence analysis, which is common in offline version of distributed optimization analysis methods, such as [18] and [20]. But it gives a bound on time accumulation from $\bar{x}(t)$ to the optimal estimate, which is unique to the online version.

Theorem 2: Suppose Assumptions 1 - 5 hold. Under Algorithm 1, it holds that

$$\frac{LD_1}{D_2}\sum_{t=1}^{T}\sum_{i=1}^{n}\|x_i(t) - \bar{x}(t)\| \leq \frac{nLCD_1}{(1-\lambda)D_2}\max_{i\in V}\|x_i(1)\| + \frac{n\lambda CD_1(M_1+M_2)^2m^2L^2}{4(1-\lambda)\beta\mu D_2 M_1 M_2 \hat{L}^2}(1+\log T)), \quad (27)$$

where all the notations are the same as given in Theorem 1.

Proof: Define

$$r_i(t) = x_i(t) - \hat{x}_i(t) = \Pi_{\mathcal{X}}^{H_i(t-1)}(\hat{x}_i(t)) - \hat{x}_i(t)$$

Since $\mathcal{X}$ is a convex set, $x_i(t) \in \mathcal{X}$ for all i, and $A(t)$ is a row stochastic matrix, therefore $\sum_{j=1}^{n} a_{ij}(t)x_j(t) \in \mathcal{X}$ for all i. By the definition of projection and lemma 2, the norm of $r_i(t)$ can be bounded

$$\|r_i(t)\| = \|\Pi_{\mathcal{X}}^{H_i(t-1)}(\hat{x}_i(t)) - \hat{x}_i(t)\| \leq \|\sum_{j=1}^{n} a_{ij}(t)x_j(t) - \hat{x}_i(t)\| \quad (28)$$

Let

$$x(t) = [x_1(t), x_2(t), \cdots, x_n(t)]^T,$$
$$\varepsilon(t) = [\frac{H_1(t)^{-1}\nabla_{t,2}}{\beta z_{11}(t)}, \frac{H_2(t)^{-1}\nabla_{t,1}}{\beta z_{22}(t)}, \cdots, \frac{H_n(t)^{-1}\nabla_{t,n}}{\beta z_{nn}(t)}]^T.$$
$$r(t) = [r_1(t), r_2(t), \cdots, r_n(t)]^T$$

Then, we can rewrite the updating formula in Algorithm 1 as the following compact form

$$x(t) = A(t:1)x(1) - \sum_{k=2}^{t} A(t:k)\varepsilon(k-1) + \sum_{k=2}^{t} A(t:k)r(k-1). \quad (29)$$

Since $|a_{ij}(t:k) - \pi_j(k)| \leq C\lambda^{t-k}$, for any $i, j \in V$ and all $t \geq k \geq 0$ by Lemma 3, and $\|r_i(t)\| \leq \|\sum_{j=1}^{n} a_{ij}(t)x_j(t) - \hat{x}_i(t)\| = \|\varepsilon_i(k)\|$, it follows from the above equation that

$$\begin{aligned}
&\|x_i(t) - \bar{x}(t)\| \\
&= \|e_i x(t) - \pi(t)^T x(t)\| \\
&= \|e_i A(t:1)x(1) - e_i\sum_{k=2}^{t} A(t:k)\varepsilon(k-1) \\
&\quad + e_i\sum_{k=2}^{t} A(t:k)r(k-1) - \pi(t)^T A(t:1)x(1) \\
&\quad + \pi(t)^T\sum_{k=2}^{t} A(t:k)\varepsilon(k-1) - \pi(t)^T\sum_{k=2}^{t} A(t:k)r(k-1)\| \\
&\leq \|(e_i A(t:1) - \pi(t)^T A(t:1))x(1)\| \\
&\quad + \|\sum_{k=2}^{t} e_i A(t:k)\varepsilon(k-1) - \sum_{k=2}^{t}\pi(t)^T A(t:k)\varepsilon(k-1)\| \\
&\quad + \|\sum_{k=2}^{t} e_i A(t:k)r(k-1) - \sum_{k=2}^{t}\pi(t)^T A(t:k)r(k-1)\| \\
&\leq C\lambda^{t-1}\max_{i\in V}\|x_i(1)\| + 2C\sum_{k=2}^{t}\lambda^{t-k}\|\varepsilon(k-1)\|. \quad (30)
\end{aligned}$$

Let ρ_i be the eigenvalue of $H_i(t)$. Then, $\frac{1}{\rho_i}$ is the eigenvalue of $H_i(t)^{-1}$. Denote $M_1 = \max_{i\in V}\rho_i, M_2 = \min_{i\in V}\rho_i$. By Schweizer inequality (Sec 2.11 in [36]) we have

$$\mathrm{tr}(H_i(t)^{-1}) \leq \frac{(M_1+M_2)^2m^2}{4M_1M_2\mathrm{tr}(H_i(t))}. \quad (31)$$

Because $H_i(t)$ is definitely positive, $H_i(t)^{-1}$ is also definitely positive. Hence we get

$$\begin{aligned}
&\|H_i(t)^{-1}\nabla_{t,i}\|^2 \\
&= \mathrm{tr}(H_i(t)^{-1})^2(\nabla_{t,i}\nabla_{t,i}^T) \\
&\leq (\mathrm{tr}(H_i(t)^{-1}))^2\mathrm{tr}(\nabla_{t,i}\nabla_{t,i}^T) \leq (\frac{(M_1+M_2)^2m^2}{4M_1M_2\mathrm{tr}(H_i(t))})^2L^2. \quad (32)
\end{aligned}$$

The last inequality in (32) holds because $\nabla_{t,i}\nabla_{t,i}^T$ is definitely positive, and hence $\mathrm{tr}(\nabla_{t,i}\nabla_{t,i}^T) = \|\nabla_{t,i}\|^2 \leq L^2$. Then, considering $\mathrm{tr}(H_i(t)) = \sum_{r=1}^{t}\|\nabla_{r,i}\|^2 + m\epsilon$, and letting $\mu = \min_{i\in V, t\geq 0} z_{ii}(t)$ we get

$$\|\varepsilon(t)\| = \|\frac{H_i(t)^{-1}\nabla_{t,i}}{\beta z_{ii}(t)}\| \leq \frac{(M_1+M_2)^2m^2L}{4M_1M_2\beta\mu(\sum_{r=1}^{t}\|\nabla_{r,i}\|^2 + m\epsilon)} \leq \frac{(M_1+M_2)^2m^2L}{4M_1M_2\beta\mu(\sum_{r=1}^{t}\|\nabla_{r,i}\|^2 + \epsilon)}. \quad (33)$$

Introducing a new positive constant $\hat{L}$ such that $\hat{L}^2 \leq \frac{1}{k-1}\sum_{r=1}^{k-1}\|\nabla_{r,i}\|^2$ implies that $(k-1)\hat{L}^2 \leq \sum_{r=1}^{k-1}\|\nabla_{r,i}\|^2 + \epsilon$. With this new notation and the notations defined by (17)

and (19), substituting (33) into (30) and summing the results on $i = 1, \cdots, n, t = 1, \cdots, T$ yields

$$\begin{aligned}
&\frac{LD_1}{D_2}\sum_{t=1}^{T}\sum_{i=1}^{n}\|x_i(t)-\bar{x}(t)\| \\
&\le \frac{LCD_1}{D_2}\sum_{i=1}^{n}\Bigg(\sum_{t=1}^{T}\lambda^{t-1}\max_{i\in V}\|x_i(1)\| \\
&\quad +2\sum_{t=1}^{T}\sum_{k=2}^{t}\lambda^{t-k}\frac{(M_1+M_2)^2m^2L}{4M_1M_2\beta\mu(\sum_{r=1}^{k-1}\|\nabla_{r,i}\|^2+\epsilon)}\Bigg) \\
&\le \frac{LCD_1}{D_2}\sum_{i=1}^{n}\Bigg(\frac{1}{1-\lambda}\max_{i\in V}\|x_i(1)\| \\
&\quad +\frac{(M_1+M_2)^2m^2L}{2\beta\mu M_1M_2\hat{L}^2}\sum_{t=1}^{T}\sum_{k=2}^{t}\lambda^{t-k}\frac{1}{k-1}\Bigg). \qquad (34)
\end{aligned}$$

Let $I(t{>}k)$ be an indicator function which takes value 1 as $t > k$, and takes 0 otherwise. Therefore,

$$\begin{aligned}
&\sum_{t=1}^{T}\sum_{k=2}^{t}\lambda^{t-k}\frac{1}{k-1}=\sum_{t=1}^{T}\sum_{k=1}^{t-1}\lambda^{t-k-1}\frac{1}{k} \\
&=\sum_{t=1}^{T}\sum_{k=1}^{T}\lambda^{t-k-1}\frac{1}{k}I(t>k)=\sum_{k=1}^{T}\frac{1}{k}\sum_{t=k+1}^{T}\lambda^{t-k-1} \\
&\le\sum_{k=1}^{T}\frac{1}{k}\sum_{t=1}^{T}\lambda^{t-1}\le\frac{1}{1-\lambda}(1+\log T). \qquad (35)
\end{aligned}$$

Substituting (35) into (34) yields

$$\begin{aligned}
\frac{LD_1}{D_2}\sum_{t=1}^{T}\sum_{i=1}^{n}\|x_i(t)-\bar{x}(t)\| &\le \frac{nLCD_1}{(1-\lambda)D_2}\max_{i\in V}\|x_i(1)\| \\
&+\frac{nCD_1(M_1+M_2)^2m^2L^2}{2(1-\lambda)\beta\mu D_2M_1M_2\hat{L}^2}(1+\log T)) \qquad (36)
\end{aligned}$$

Then, the proof is completed.

In order to accomplish the proof of our next theorem, we need to prove the following lemma at first.

Remark 5: Theorem 2 establishes the consensus part of convergence analysis. If the projection error is bounded, then the difference between the estimates of any two different agents in the network has an upper bound and gradually converges over time.

Lemma 4: Suppose Assumptions 1 - 5 hold. Consider Algorithm 1. With the notation defined in (18) and (19), it holds that

$$\frac{D_3}{2\beta D_2}\sum_{t=1}^{T}\sum_{i=1}^{n}\nabla_{t,i}^{T}H_i(t)^{-1}\nabla_{t,i}\le\frac{nmD_3}{2\beta D_2}\log(\frac{TL^2}{\epsilon}+1). \qquad (37)$$

Proof: For matrices $A, B \in \mathbb{R}^{m\times m}$, denote $A\bullet B = \sum_{i=1}^{m}\sum_{j=1}^{m}A_{ij}B_{ij} = \mathrm{tr}(AB^T)$, which can be considered as the inner product of matrices A and B if these matrices are considered as vectors in the space $\mathbb{R}^{m^2}$. By Lemma 4.6 in [4], it holds that $A^{-1}\bullet(A-B)\le\log\frac{|A|}{|B|}$, where $|A|$ denotes the determinant of matrix A. Using this fact, we get

$$\begin{aligned}
\sum_{t=1}^{T}\nabla_{t,i}^{T}H_i(t)^{-1}\nabla_{t,i} &= \sum_{t=1}^{T}H_i(t)^{-1}\bullet(H_i(t)-H_i(t-1)) \\
&\le\sum_{t=1}^{T}\log\frac{|H_i(t)|}{|H_i(t-1)|}=\log\frac{|H_i(T)|}{|H_i(0)|}. \qquad (38)
\end{aligned}$$

Since $H_i(t)=\sum_{r=1}^{t}\nabla_{r,i}\nabla_{r,i}^{T}+\epsilon I_m$, $\|\nabla_{t,i}\|\le L$, it follows that the maximum eigenvalue of $H_i(T)$ is not greater than $TL^2+\epsilon$. Therefore, $|H_i(T)|\le (TL^2+\epsilon)^m$, $|H_i(0)|=\epsilon^m$. Substituting this equality into (38) yields

$$\begin{aligned}
&\frac{D_3}{2\beta D_2}\sum_{t=1}^{T}\sum_{i=1}^{n}\nabla_{t,i}^{T}H_i(t)^{-1}\nabla_{t,i} \\
&\le\frac{D_3}{2\beta D_2}\sum_{i=1}^{n}\log\frac{|H_i(T)|}{|H_i(0)|} \\
&\le\frac{D_3}{2\beta D_2}\sum_{i=1}^{n}m\log(\frac{TL^2}{\epsilon}+1)=\frac{nmD_3}{2\beta D_2}\log(\frac{TL^2}{\epsilon}+1).
\end{aligned}$$

Remark 6: Lemma 4 gives the upper bound of the third term in (26), and provides supporting conditions for the final convergence result. It is different from first-order methods such as online gradient descent in [9] and [17], because it estimates the Hessian matrix.

Now we are at a position to present our main result.

Theorem 3: Suppose Assumptions 1 - 5 hold. Then, Algorithm 1 ensures

$$\begin{aligned}
R_T &= \sum_{t=1}^{T}\sum_{i=1}^{n}(f_{t,i}(x_i(t))-f_{t,i}(x^*)) \\
&\le(\tilde{C}_1+\tilde{C}_2+\tilde{C}_3)+(\tilde{C}_3+\tilde{C}_4)\log T \qquad (39)
\end{aligned}$$

where

$$\begin{aligned}
\tilde{C}_1 &= \frac{\beta n\epsilon D^2}{2D_2}, \\
\tilde{C}_2 &= \frac{nL(CD_1+D_2)}{(1-\lambda)D_2}\max_{i\in V}\|x_i(1)\|, \\
\tilde{C}_3 &= \frac{n(CD_1+D_2)(M_1+M_2)^2m^2L^2}{2(1-\lambda)\beta\mu D_2M_1M_2\hat{L}^2}, \\
\tilde{C}_4 &= \frac{nmD_3}{2\beta D_2}
\end{aligned}$$

with D_1, D_2 and D_3 being defined in (17), (18) and (19), respectively.

Proof: Substituting (27) and (37) into (5) yields

$$\begin{aligned}
&\sum_{t=1}^{T}(F_t(\bar{x}(t))-F_t(x^*)) \\
&\le\frac{\beta n\epsilon D^2}{2D_2}+\frac{nLCD_1}{(1-\lambda)D_2}\max_{i\in V}\|x_i(1)\|+\frac{nmD_3}{2\beta D_2}\log(\frac{TL^2}{\epsilon}+1)
\end{aligned}$$

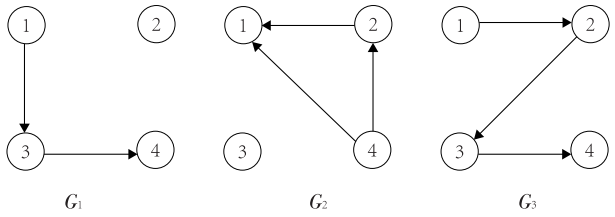

FIGURE 1. Time-varing unbalanced digraph used in simulation.

$$+\frac{nCD_1(M_1+M_2)^2m^2L^2}{2(1-\lambda)\beta\mu D_2M_1M_2\hat{L}^2}(1+\log T). \tag{40}$$

Using (27) we have

$$\begin{aligned}
&\sum_{t=1}^{T}\sum_{i=1}^{n}(f_{t,i}(x_i(t))-f_{t,i}(\bar{x}(t)))\\
&\le \sum_{t=1}^{T}\sum_{i=1}^{n}(L\|x_i(t)-\bar{x}(t)\|)\\
&\le \frac{nL}{(1-\lambda)}\max_{i\in V}\|x_i(1)\|+\frac{n(M_1+M_2)^2m^2L^2}{2(1-\lambda)\beta\mu M_1M_2\hat{L}^2}(1+\log T).
\end{aligned} \tag{41}$$

Substituting (40) and (41) into the following equation, we get

$$\begin{aligned}
R_T &= \sum_{t=1}^{T}\sum_{i=1}^{n}(f_{t,i}(x_i(t))-f_{t,i}(x^*))\\
&= \sum_{t=1}^{T}\sum_{i=1}^{n}(f_{t,i}(x_i(t))-f_{t,i}(\bar{x}(t))+f_{t,i}(\bar{x}(t))-f_{t,i}(x^*))\\
&\le \frac{\beta n\epsilon D^2}{2D_2}+\frac{nL(CD_1+D_2)}{(1-\lambda)D_2}\max_{i\in V}\|x_i(1)\|\\
&\quad+\frac{n(CD_1+D_2)(M_1+M_2)^2m^2L^2}{2(1-\lambda)\beta\mu D_2M_1M_2\hat{L}^2}(1+\log T)\\
&\quad+\frac{nmD_3}{2\beta D_2}\log(\frac{TL^2}{\epsilon}+1).
\end{aligned} \tag{42}$$

Recall that $\epsilon = \frac{1}{\beta^2D^2}$, and $\beta = \frac{1}{2}\min\{\frac{1}{4LD},\alpha\}$. Then, it follows that $\log(\frac{TL^2}{\epsilon}+1)\le\log(\frac{T}{64}+1)\le\log T$ (for $T>1$). Hence, (42) implies (39). The proof is completed.

Remark 7: Theorem 3 gives the main result of algorithm 1, namely, regret converges sublinearly with logarithmic velocity. This shows that when T is large enough, our algorithm can achieve the same good results as the offline optimization method under time-varying unbalanced graphs.

IV. NUMERICAL EXAMPLES

In this section we provide two examples to illustrate the advantages of our online algorithm over the existing ones from different aspects.

The first example considers the distributed estimation problem in sensor networks, which has been considered in [10]. The sensors are required to collaboratively estimate some unknown parameter vector (position of a target, for example) denoted as $x\in\mathbb{R}^d$. Each sensor assumes a linear parametric model H_ix to model its observation variable $z_i\in\mathbb{R}^p$, where $H_i\in\mathbb{R}^{p\times d}$ is the information structure matrix. At each moment $t\in\{1,\dots,T\}$, each sensor will obtain an actual observation which is generated by a time-varying function $z_i(t)=a_i(t)x+b_i(t)$, where $a_i(t),b_i(t)$ reflects the environment variation and observation noise, respectively. The goal is to find the estimate x to minimize the following loss function

$$f_t(x)=\sum_{t=1}^{T}\sum_{i=1}^{n}\frac{1}{2}\|z_i(t)-H_ix\|_2^2.$$

Since the time-varying factors $a_i(t),b_i(t)$ are unknown to agents only afterwards, online estimation algorithms are suitable for this problem.

Suppose there are four sensors in the network. The communication topology of sensors is time-varying and

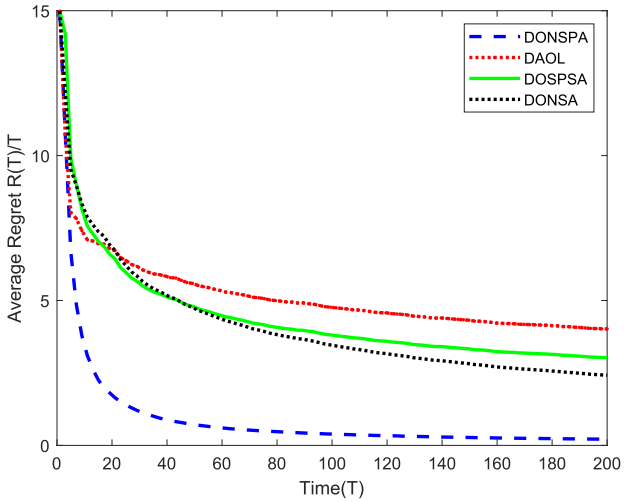

FIGURE 2. Average regret trends for the four algorithms in example 1.

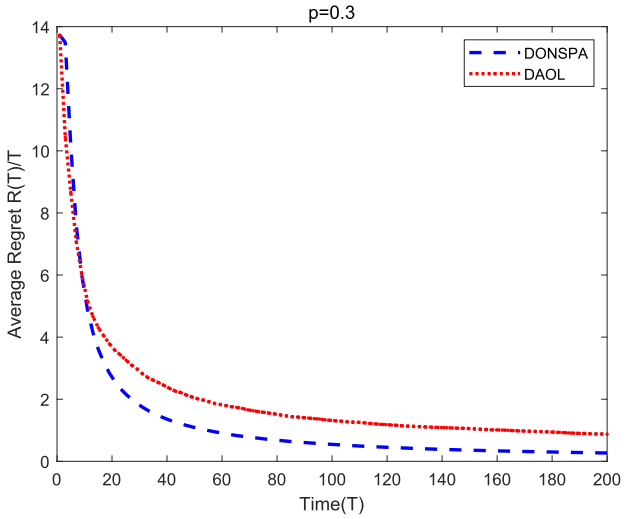

FIGURE 3. Comparison of average regrets when the switching probability of G_2 is 0.3.

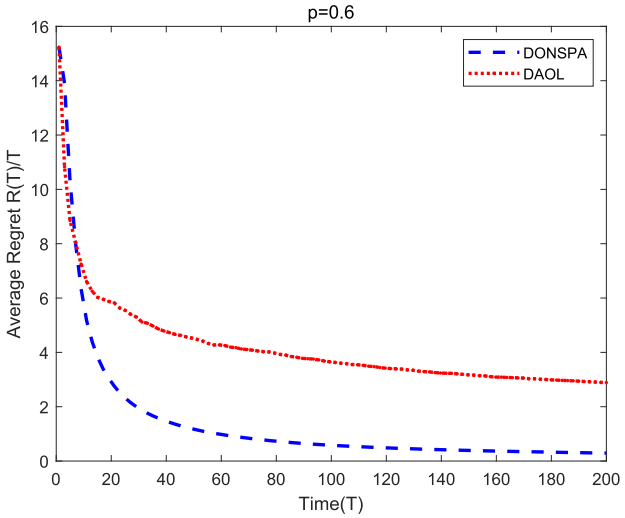

FIGURE 4. Comparison of average regrets when the switching probability of G_2 is 0.6.

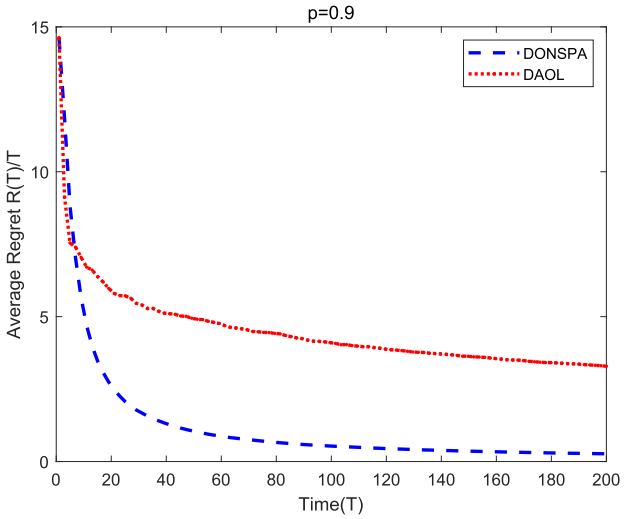

FIGURE 5. Comparison of average regrets when the switching probability of G_2 is 0.9.

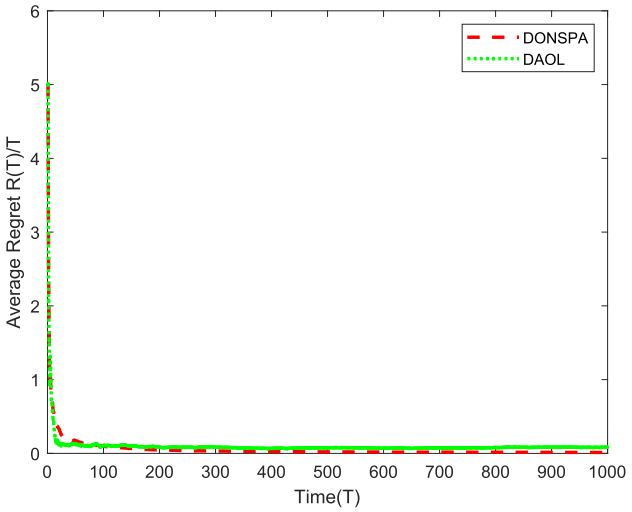

FIGURE 6. Average regret of two algorithms in Example 2.

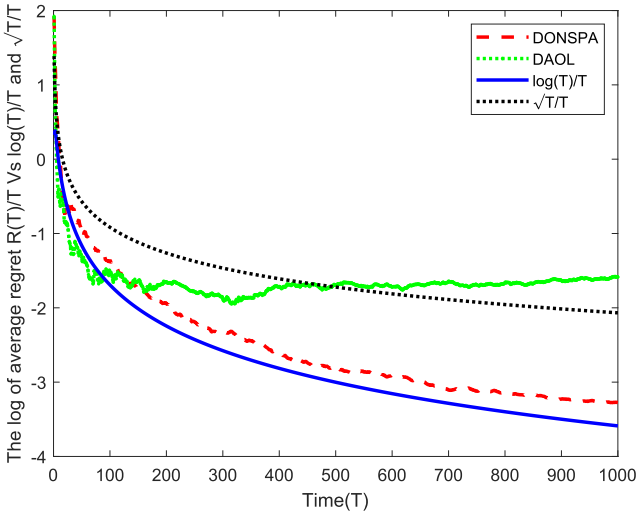

FIGURE 7. Convergence speed of two algorithms in example 2.

switching in turn among three unbalanced digraphs shown in Fig. 1. Obviously, the joint graph is strongly connected. We assume the dimensions of the observation vector and the parameter vector are both equal to one, i.e., $p = d = 1$, and $H_i = 1/i, i = 1, 2, 3, 4$. We also assume for each i, $a_i(t)$ and $b_i(t)$ are chosen randomly from a uniform distribution on $[0, 2]$ and $[-\frac{1}{2}, \frac{1}{2}]$, respectively.

We apply four different algorithms, namely the distributed online Newton step projection algorithm (DONSPA) proposed in this paper, the distributed autonomous online learning algorithm (DAOL) proposed in [9], the distributed online subgradient push-sum algorithm (DOSPSA) proposed in [17] and the distributed online Newton step algorithm (DONSA) proposed in [32] to the estimation. Thanks to the strongly convexity of the loss function, all three algorithms theoretically ensure logarithmic convergence of the regret bound. However, Fig. 2 shows the evident difference between our result and the results of DAOL, DOSPSA and DONSA. More specifically, the average regret of DONSPA, DAOL, DOSPA and DONSA is 0.36, 4.10 and 3.09 and 2.41 respectively at the 200th iteration. Our algorithm is 8.77%, 11.62% and 14.94% of the other two, respectively. And this difference does not disappear even after 200 iterations. The reason is that each agent has different "optimal" estimate, and the optimal estimate defined by the sum-type loss function can be obtained only based on their balanced cooperation. However, in this example the communication digraph is unbalanced, and among three algorithms only our algorithm can compensate for this unbalancedness.

In order to better illustrate the influence of unbalanced topology on mean regret convergence, we set the probabilities of the second topology in Figure 1, which are 0.3, 0.6, and 0.9, respectively, with other conditions being the same. We compared the average regrets of DONSPA and DAOL

under different probabilities. As can be seen from Fig. 3-5, The average regret of DAOL varies with the appearance of G2. Our algorithm compensates for the weight, so it keeps the average regret small in all cases.

Now, let us consider the second example to see the convergence speed of different algorithms for non-strongly convex loss functions. The communication digraph of four agents here is the same as used in the first one. But the loss function of each agent is replaced by $f_{i,t}(x) = -\log(r_i(t)^T x)$, which is often used in portfolio management [4]. This function is not strongly convex, but just exp-concave.

Now, we compare the convergence speed of our algorithm DONSPA and the first-order algorithm DAOL. It can be seen from the results in Fig. 6 that the average regret of the two algorithms will decrease with time, but it is difficult to see the difference in convergence speed between them. To show their convergence speeds more clearly, we take the logarithmic value of the average regret as the vertical coordinate, and depict two standard curves, namely $\frac{\log T}{T}$ and $\frac{\sqrt{T}}{T}$ together with the regret bounds. Simulation results given in Fig. 7 show that DONSPA is approaching to the logarithm curve, while DAOL is going to the square root curve. The results are consistent with our theoretical analysis and the conclusion in [9].

V. CONCLUSION

In this paper, distributed online Newton step algorithm is extended to the case of time-varying unbalanced network. Theoretical analysis shows that our algorithm can make the decision value of each node tend to the optimal value in hindsight in the time-varying general digraphs, that is, the regret bound converges sublinearly with respect to the number of iterations, and achieves logarithm convergence speed. It does not need use double stochastic matrix when exchanging information, which greatly expands the application range of the original algorithm. Numerical examples demonstrate the effectiveness and advantages of the proposed algorithm. Future research interests include, consider extending the existing algorithm to the case with inequality constraints.

ACKNOWLEDGMENT

The authors would like to thank Dr. Yanqiong Zhang of the School of Automation, Hangzhou Dianzi University, for helpful discussion.

REFERENCES

[1] T. Cover, "Universal portfolis," *Math. Finance*, vol. 1, no. 1, pp. 1–19, 1991.
[2] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *Proc. 20th Int. Conf. Mach. Learn. (ICML)*, 2003, pp. 928–936.
[3] E. Hazan, A. Agarwal, and S. Kale, "Logarithmic regret algorithms for online convex optimization," *Mach. Learn.*, vol. 69, nos. 2–3, pp. 169–192, Oct. 2007.
[4] E. Hazan, "Introduction to online convex optimization," *Found. Trends Optim.*, vol. 2, nos. 3–4, pp. 157–325, 2016.
[5] S. Hosseini, A. Chapman, and M. Mesbahi, "Online distributed optimization via dual averaging," in *Proc. 52nd IEEE Conf. Decis. Control*, Dec. 2013, pp. 1484–1489.
[6] Q. Ling and A. Ribeiro, "Decentralized dynamic optimization through the alternating direction method of multipliers," *IEEE Trans. Signal Process.*, vol. 62, no. 5, pp. 1185–1197, Mar. 2014.
[7] S. S. Ram, A. Nedić, and V. V. Veeravalli, "Distributed stochastic subgradient projection algorithms for convex optimization," *J. Optim. Theory Appl.*, vol. 147, no. 3, pp. 516–545, Dec. 2010.
[8] S. Shalev-Shwartz, "Online learning and online convex optimization," *Found. Trends Mach. Learn.*, vol. 4, no. 2, pp. 107–194, 2011.
[9] F. Yan, S. Sundaram, S. V. N. Vishwanathan, and Y. Qi, "Distributed autonomous online learning: Regrets and intrinsic privacy-preserving properties," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 11, pp. 2483–2493, Nov. 2013.
[10] S. Hosseini, A. Chapman, and M. Mesbahi, "Online distributed convex optimization on dynamic networks," *IEEE Trans. Autom. Control*, vol. 61, no. 11, pp. 3545–3550, Nov. 2016.
[11] B. Gharesifard and J. Cortés, "Distributed continuous-time convex optimization on weight-balanced digraphs," *IEEE Trans. Autom. Control*, vol. 59, no. 3, pp. 781–786, Mar. 2014.
[12] D. Mateos-Núñez and J. Cortés, "Distributed online convex optimization over jointly connected digraphs," *IEEE Trans. Netw. Sci. Eng.*, vol. 1, no. 1, pp. 23–37, Jan. 2014.
[13] K. Lu, G. Jing, and L. Wang, "Online distributed optimization with strongly pseudoconvex-sum cost functions," *IEEE Trans. Autom. Control*, vol. 65, no. 1, pp. 426–433, Jan. 2020.
[14] X. Li, X. Yi, and L. Xie, "Distributed online convex optimization with an aggregative variable," *IEEE Trans. Control Netw. Syst.*, vol. 9, no. 1, pp. 438–449, Mar. 2022.
[15] H. Li and Y. Shi, "Network-based predictive control for constrained nonlinear systems with two-channel packet dropouts," *IEEE Trans. Ind. Electron.*, vol. 61, no. 3, pp. 1574–1582, Mar. 2014.
[16] X. Wu, Y. Tang, and W. Zhang, "Stability analysis of stochastic delayed systems with an application to multi-agent systems," *IEEE Trans. Autom. Control*, vol. 61, no. 12, pp. 4143–4149, Dec. 2016.
[17] M. Akbari, B. Gharesifard, and T. Linder, "Distributed online convex optimization on time-varying directed graphs," *IEEE Trans. Control Netw. Syst.*, vol. 4, no. 3, pp. 417–428, Sep. 2017.
[18] A. Nedic and A. Olshevsky, "Distributed optimization over time-varying directed graphs," *IEEE Trans. Autom. Control*, vol. 60, no. 3, pp. 601–615, Mar. 2015.
[19] X. Li, X. Yi, and L. Xie, "Distributed online optimization for multi-agent networks with coupled inequality constraints," *IEEE Trans. Autom. Control*, vol. 66, no. 8, pp. 3575–3591, Aug. 2021.
[20] V. S. Mai and E. H. Abed, "Distributed optimization over weighted directed graphs using row stochastic matrix," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2016, pp. 7165–7170.
[21] H. Li, Q. Lü, and T. Huang, "Distributed projection subgradient algorithm over time-varying general unbalanced directed graphs," *IEEE Trans. Autom. Control*, vol. 64, no. 3, pp. 1309–1316, Mar. 2019.
[22] Y. Xiong, J. Xu, K. You, J. Liu, and L. Wu, "Privacy-preserving distributed online optimization over unbalanced digraphs via subgradient rescaling," *IEEE Trans. Control Netw. Syst.*, vol. 7, no. 3, pp. 1366–1378, Sep. 2020.
[23] K. Tada, N. Hayashi, and S. Takai, "Distributed inequality constrained online optimization for unbalanced digraphs using row stochastic property," in *Proc. IEEE 61st Conf. Decis. Control (CDC)*, Dec. 2022, pp. 2283–2288.
[24] J. Li, C. Li, W. Yu, X. Zhu, and X. Yu, "Distributed online bandit learning in dynamic environments over unbalanced digraphs," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 4, pp. 3034–3047, Oct. 2021.
[25] J. Li, X. Zhu, Z. Wu, and T. Huang, "Online distributed dual averaging algorithm for multi-agent bandit optimization over time-varying general directed networks," *Inf. Sci.*, vol. 581, pp. 678–693, Dec. 2021.
[26] Y. Pang and G. Hu, "Randomized gradient-free distributed online optimization with time-varying cost functions," in *Proc. IEEE 58th Conf. Decis. Control (CDC)*, Dec. 2019, pp. 4910–4915.
[27] C. Xi and U. A. Khan, "Distributed subgradient projection algorithm over directed graphs," *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 3986–3992, Aug. 2017.
[28] Y. Xiong, X. Li, K. You, and L. Wu, "Distributed online optimization in time-varying unbalanced networks without explicit subgradients," *IEEE Trans. Signal Process.*, vol. 70, pp. 4047–4060, 2022.
[29] A. Mokhtari, Q. Ling, and A. Ribeiro, "An approximate Newton method for distributed optimization," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 2959–2963.

[30] D. Bajović, D. Jakovetić, N. Krejić, and N. K. Jerinkić, "Newton-like method with diagonal correction for distributed optimization," *SIAM J. Optim.*, vol. 27, no. 2, pp. 1171–1203, Jan. 2017.
[31] M. Wu, N. Xiong, A. V. Vasilakos, V. C. M. Leung, and C. L. P. Chen, "RNN-K: A reinforced Newton method for consensus-based distributed optimization and control over multiagent systems," *IEEE Trans. Cybern.*, vol. 52, no. 5, pp. 4012–4026, May 2022.
[32] X. Chu, "A distributed online Newton step algorithm for multi-agent systems," *Math. Problems Eng.*, vol. 2022, pp. 1–14, Oct. 2022.
[33] K. S. Riedel, "A Sherman–Morrison–Woodbury identity for rank augmenting matrices with application to centering," *SIAM J. Matrix Anal. Appl.*, vol. 13, no. 2, pp. 659–662, Apr. 1992.
[34] A. Nedic, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Trans. Autom. Control*, vol. 55, no. 4, pp. 922–938, Apr. 2010.
[35] A. Nedic and J. Liu, "On convergence rate of weighted-averaging dynamics for consensus problems," *IEEE Trans. Autom. Control*, vol. 62, no. 2, pp. 766–781, Feb. 2017.
[36] D. S. Mitrinovic and P. M. Vasic, *Analytic Inequalities*, vol. 1. Berlin, Germany: Springer, 1970.

JIAYI WU received the B.S. degree in electrical engineering and automation from Hangzhou Dianzi University, Hangzhou, Zhejiang, China, in 2021, where she is currently pursuing the M.S. degree in control science and engineering. Her research interests include distributed cooperative control of multi-agent systems, distributed optimization, and game theory.

YU-PING TIAN received the bachelor's degree in automation from Tsinghua University, Beijing, China, in 1986, the Ph.D. degree in electrical engineering from the Moscow Power Engineering Institute, Moscow, Russia, in 1991, and the D.Sc. degree in electrical engineering from the Taganrog State University of Radioengineering, Taganrog, Russia, in 1996. From 1992 to 2018, he was with the School of Automation, Southeast University, Nanjing, China. He held a visiting position with Central Queensland University, Rockhampton, QLD, Australia, the University of California at Berkeley, Berkeley, CA, USA, and the City University of Hong Kong, Hong Kong. He is currently a Chair Professor with the School of Automation, Hangzhou Dianzi University, Hangzhou, China. His research interests include distributed control, estimation, learning and inference theory and applications to wireless sensor networks, and multi-robot systems. He was a recipient of the Chang Jiang Distinguished Professorship awarded by the Education Ministry of China and the Distinguished Young Scholar Award by the National Natural Science Foundation of China.