

(Rectified Version) The Barzilai-Borwein Method for Distributed Optimization over Unbalanced Directed Networks

Jinhui Hu^a, Xin Chen^{b,*}, Lifeng Zheng^a, Ling Zhang^a, Huaqing Li^{a,*}

^a*Chongqing Key Laboratory of Nonlinear Circuits and Intelligent Information Processing, College of Electronic and Information Engineering, Southwest University, Chongqing 400715, P.R. China*

^b*School of Big Data & Software Engineering, Chongqing University, Chongqing, 401331 P.R. China.*

Abstract

This paper studies optimization problems over multi-agent systems, in which all agents cooperatively minimize a global objective function expressed as a sum of local cost functions. Each agent in the systems uses only local computation and communication in the overall process without leaking their private information. Based on the Barzilai-Borwein (BB) method and multi-consensus inner loops, a distributed algorithm with the availability of larger step-sizes and accelerated convergence, namely ADBB, is proposed. Moreover, owing to employing only row-stochastic weight matrices, ADBB can resolve the optimization problems over unbalanced directed networks without requiring the knowledge of neighbors' out-degree for each agent. Via establishing contraction relationships between the consensus error, the optimality gap, and the gradient tracking error, ADBB is theoretically proved to converge linearly to the globally optimal solution. A real-world data set is used in simulations to validate the correctness of the theoretical analysis.

Keywords: Multi-agent systems; distributed optimization; BB method; multi-step communications; high-performance algorithms.

*Corresponding author
Email addresses: richard_chen@126.com (Xin Chen), huaqingli@swu.edu.cn (Huaqing Li)

1. Introduction

Distributed optimization is a promising paradigm that finds many practical applications such as signal processing [1], machine learning [2, 3, 4], coordinated control [5, 6], resource allocation [7], deep learning [8, 9], and Internet of Things [10]. Clearly, distributed optimization is robust than traditional centralized optimization in terms of communication networks and various practical problems can be modeled as distributed optimization, where m agents cooperatively resolve the following optimization problem

$$\min_{\tilde{x} \in \mathbb{R}^n} f(\tilde{x}) = \frac{1}{m} \sum_{i=1}^m f_i(\tilde{x}), \quad (1)$$

where each agent in the system has the knowledge of only one local objective function, $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$. Furthermore, each agent can only receive information from its in-neighbors and transmit information to its out-neighbors. The mutual goal of all agents in the network is to seek the optimal solution, \tilde{x}^* , of problem (1) through communicating with its neighbors with no leak of their private information.

1.1. Literature Review

There is a large amount of outstanding works concerning with distributed optimization methods, including the distributed (sub)gradient method [11], the distributed primal-dual (sub)gradient method [12], the distributed augmented Lagrangian method [13], and the Newton method [14]. All distributed optimization methods can be summarized into two categories. One is Lagrangian dual variables based methods, including distributed dual decomposition [15], distributed alternating direction method of multipliers (ADMM) [16, 17, 18]. These algorithms have the superiority of achieving exact globally optimal solution while suffering from more computational complexity than primal methods. Especially, [17, 18] are proved to converge linearly to the globally optimal solution under the conditions that the objective functions are strongly convex and have Lipschitz gradients. Some other existing methods like, DIGing [19] and

EXTRA [20], can reach the globally optimal solution by using a sufficiently small constant step-size with no dual variables updating explicitly, which can be considered as an augmented Lagrangian primal-dual methods with a single gradient step in primal space. Another category is first-order primal methods, which are well developed in recent years owing to its simplicity and high efficiency. Some early notable methods include (sub)gradient descent (DGD) based methods [21, 22]. In these algorithms, each agent in the system updates its local estimate according to a combination of a predefined consensus step and a local gradient descent step.

However, the common drawback of the above methods is the requirement of constructing various doubly stochastic weight matrices, which demands undirected or balanced directed networks between agents. This requirement may significantly restrict the applicability of these methods in practical applications because all agents in the network may broadcast at diverse power levels which indicates the communication capability in one direction while not in the other [23]. Therefore, some researchers are devoted to studying the algorithms over unbalanced directed networks. Based on distributed (sub)gradient descent (DGD) [21, 22], Xi *et al.* in [11, 24] leverage a so-called surplus-based method to realize exact convergence. Then, the (sub)gradient-push algorithm [25] that can be applied to unbalanced directed networks constructs only column-stochastic weight matrices to achieve the globally optimal solution via incorporating a push-sum technique [26] into DGD-based methods [21, 22]. However, the method [25] shows relatively slow convergence rate due to the use of diminishing step-sizes. So as to accelerate the convergence, DEXTRA [27] combines the push-sum technique with EXTRA [20] to achieve linear convergence under the standard strong convexity assumption with the step-size lying in some non-trivial interval. The restriction on step-size is relaxed by the follow-up works ADD-OPT/Push-DIGing [26, 19], \mathcal{AB} [28], and $\mathcal{AB}m$ [29], where \mathcal{AB} simultaneously utilizes both row- and column-stochastic weight matrices to gain an accelerated convergence rate. Based on \mathcal{AB} , $\mathcal{AB}m$ first introduces distributed heavy-ball type acceleration into distributed optimization. Notice that these algorithms

[24, 26, 28, 30, 31] can better protect from using the doubly stochastic weight matrices and can be applied into unbalanced directed networks. However, all these methods may be impractical to some realistic environment that all agents in the networks adhere to a broadcast-based communication protocol, i.e, the agents in the network only get the knowledge of its in-degree while do not know its out-degree. Therefore, Xi *et al.* in [23] propose an elegant method which is based on a gradient tracking technique [22] and uses only row-stochastic weight matrices to achieve exact convergence. FROST [30] and [32] extend [23] by employing uncoordinated step-sizes, which further relax the restriction on step-sizes. Theoretically, the step-sizes of FROST with linear convergence rate in the strongly convex case do not exceed $(1/mL_f)$ where m is the number of agents and L_f is the Lipschitz continuous constant. Notice that the upper bound $(1/mL_f)$ on step-sizes may be very small, which may limit the convergence rate of FROST. Thus, an useful method named Barzilai Borwein (BB) method is introduced into distributed optimization over undirected networks by DGM-BB-C [33]. In [33], Gao *et al.* conduct multi-consensus inner loops to ensure both larger step-sizes and a network-independent range of step-sizes, which not only attains faster convergence than most existing works, but gives a relatively simple way of choosing the step-sizes. However, DGM-BB-C [33] can only be applied into undirected or balanced directed networks due to doubly stochastic weight matrices, which may be impractical in practice.

1.2. Motivations

In recent literature [23, 26, 24, 28, 29, 30], the notion of sufficiently small step-sizes needs to be added in theoretical results for which the agents can work in a fully distributed manner. However, researchers can always set relatively larger step-sizes in simulations, which are not in line with the theoretical results. This paper aims to remove the notion of sufficiently small step-sizes over unbalanced directed networks and develops a distributed algorithm that converges to the globally optimal solution with fewer computational costs, communication costs and accelerated convergence.

1.3. Contributions

The main contributions of this paper can be summarized as follows:

(1) A novel accelerated distributed algorithm over unbalanced directed networks constructing only row-stochastic weight matrices and using BB step-sizes, termed as ADBB, is developed to solve convex optimization problems. More significantly, ADBB is based on adapt-then-combine variation of [23] and FROST [30], and uses multi-consensus inner loops to simultaneously ensure larger step-sizes and accelerated convergence. Additionally, ADBB is theoretically demonstrated to converge to the globally optimal solution when the local objective functions are smooth and strongly convex.

(2) Compared with some recent well-known works [20, 23, 26, 27, 28, 29, 31, 30], ADBB further eliminates the restriction on step-sizes via using BB method which witnesses many successful applications in the fields of non-negative matrix factorization[34], non-smooth optimization [35], and machine learning [36, 37]. Especially, ADBB allows all agents in the network to independently automatically compute their step-sizes according to its local information, which not only relaxes the selection range of the step-sizes but also prevents from the heterogeneity problems [30] existing in [31, 38, 39].

(3) Unlike the research [11, 20, 24, 33, 40], we consider more realistic situations, i.e., the communication networks between agents are unbalanced directed, and particularly agents exchange information in a broadcast-based directed communication network. ADBB achieves the globally optimal solution by constructing only row-stochastic weight matrices which are much easier to implement in a distributed fashion as each agent can locally decide the weights [23, 41].

(4) In fact, some well-known algorithms, for example, ADD-OPT/PUSH-DIGing [26, 19], FROST [30], and [23, 32] converge under the requirement of the largest step-sizes no exceeding $1/mL_f$ (possibly smaller) while ADBB increases the upper bound on the largest step-size to $1/m\mu$. Intuitively, ADBB has more communication at each iterations. However, owing to the use of BB step-sizes and multi-consensus inner loops, ADBB converges to the globally optimal so-

lution with the smaller number of iterations, fewer gradient-computation costs, and fewer rounds of communication than most existing works [24, 26, 28, 29, 30], which is shown in Section 5.

1.4. Organization

The following organization of this paper is presented in this section. Preliminaries are presented in Section 2. Section 3 gives the development of ADBB. The convergence of ADBB is analyzed in Section 4. Section 5 conducts numerical experiments to resolve machine learning problems to verify the theoretical analysis. Finally, we draw a conclusion and state our future work in Section 6.

1.5. Basic Notations

In this section, we present some basic notations which are useful throughout this paper. Notice that all vectors in this paper are recognized as column vectors if no otherwise specified. The detailed definitions are given in Table 1. Based on Table 1, we further introduce the Perron-vector-weighted Euclidean norm and its induced matrix norm as follows: for arbitrary $\tilde{x} \in \mathbb{R}^m$ and $X \in \mathbb{R}^{m \times m}$

$$\begin{aligned} \|\tilde{x}\|_\pi &:= \sqrt{[\pi]_1([\tilde{x}]_1)^2 + [\pi]_2([\tilde{x}]_2)^2 + \cdots + [\pi]_m([\tilde{x}]_m)^2} = \|\text{diag}\{\sqrt{\pi}\}\tilde{x}\|_2, \\ \|X\|_\pi &:= \left\| \text{diag}\{\sqrt{\pi}\} X (\text{diag}\{\sqrt{\pi}\})^{-1} \right\|_2, \end{aligned}$$

which means $\|\tilde{x}\|_\pi \leq \bar{\pi}^{0.5}\|\tilde{x}\|_2$ and $\|\tilde{x}\|_2 \leq \underline{\pi}^{-0.5}\|\tilde{x}\|_\pi$ (see [42] for details). Denote $\vartheta := \bar{\pi}/\underline{\pi} > 1$.

2. Preliminaries

2.1. Communication Network Model

Consider m agents exchanging information with each other over an unbalanced directed network, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, 2, \dots, m\}$ is the set of agents and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the collected ordered pairs. The weighted matrix associated with the unbalanced directed network \mathcal{G} is denoted by non-negative matrix,

$A = [a_{ij}] \in \mathbb{R}^{m \times m}$ such that $a_{ij} > 0$ if $(j, i) \in \mathcal{E}$ and $a_{ij} = 0$ otherwise, where the weights a_{ij} satisfy conditions as follows:

$$a_{ij} = \begin{cases} > 0, & j \in \mathcal{N}_i^{\text{in}} \\ 0, & \text{otherwise} \end{cases}, \quad \sum_{j=1}^m a_{ij} = 1, \forall i \in \mathcal{V}. \quad (2)$$

Specifically, for arbitrary two agents, $i, j \in \mathcal{V}$, if $(j, i) \in \mathcal{E}$, then agent j can transmit information to agent i . The in-neighbors of agent i is denoted as $\mathcal{N}_i^{\text{in}}$, i.e., the set of agents can transmit messages to agent i . Correspondingly, the out-neighbors of agent i is denoted as $\mathcal{N}_i^{\text{out}}$, i.e., the set of agents can receive information from agent i . The network \mathcal{G} is considered to be balanced if $\sum_{j \in \mathcal{N}_i^{\text{out}}} a_{ji} = \sum_{j \in \mathcal{N}_i^{\text{in}}} a_{ij}$, $i \in \mathcal{V}$, and unbalanced otherwise. Both $\mathcal{N}_i^{\text{in}}$ and $\mathcal{N}_i^{\text{out}}$ include agent i .

Assumption 1. ([23, Assumption 1]) *The unbalanced directed network, \mathcal{G} , is strongly connected and each agent in the network has a unique identifier $i = 1, \dots, m$.*

Remark 1. *Notice that Assumption 1 is standard in recent literature [23, 30, 32, 43]. The strongly-connected assumption on unbalanced directed networks guarantees the state averaging of the whole multi-agent system and avoids the presence of the isolated agents.*

2.2. Problem Reformulation

In this section, an equivalent form of problem (1) is given to help conducting the following convergence analysis.

$$\begin{aligned} \min_{x \in \mathbb{R}^{mn}} f(x) &= \frac{1}{m} \sum_{i=1}^m f_i(x^i), \\ \text{s.t. } x^i &= x^j, (i, j) \in \mathcal{E}, \end{aligned} \quad (3)$$

Assumption 2. (*Smoothness*): *Each differentiable local objective function, f_i , has Lipschitz continuous gradients, i.e., for arbitrary $x, y \in \mathbb{R}^n$, it holds that*

$$\|\nabla f_i(x) - \nabla f_i(y)\|_2 \leq L_f \|x - y\|_2, \quad (4)$$

where $L_f > 0$.

Symbols	Definitions
I_n	the $n \times n$ identity matrix
$\mathbf{1}_m$	an m -dimensional column vector of all ones
e_i	an n -dimensional vector of all 0's except 1 at the i -th entry
π	the left Perron eigenvector of primitive row-stochastic matrix A
x^\top	transpose of vector x
A^\top	transpose of matrix A
A^{ij}	the (i, j) -th element of matrix A
$[x]_i$	the i -th element of vector x
$\text{diag}\{x\}$	A diagonal matrix with all the elements of vector x laying on its main diagonal
$X \leq Y$	each element in $Y - X$ is non-negative, where X and Y are two vectors or matrices
$X \otimes Y$	the Kronecker product of matrices X and Y
$\rho(X)$	the spectral radius for matrix X
$\ \cdot\ _2$	the Euclidean norm for vectors and the spectral norm for matrices

Table 1: Basic notations.

Assumption 3. (Strong convexity): Each local objective function, f_i , is strongly convex, i.e., for arbitrary $x, y \in \mathbb{R}^n$, it holds that

$$f_i(x) - f_i(y) \geq \langle \nabla f_i(y), x - y \rangle + \frac{\mu}{2} \|x - y\|_2^2, \quad (5)$$

where $\mu > 0$.

Remark 2. Note that constants L_f, μ in Assumptions 2-3 satisfy $0 < \mu \leq L_f$ (see [44]). Clearly, under Assumptions 3, original problem (1) has a unique optimal solution which is denoted as $\tilde{x}^* \in \mathbb{R}^n$. Therefore, the globally optimal solution denoted as $x^* = 1_m \otimes \tilde{x}^*$ to transformed problem (3) also exists uniquely. We emphasize that Assumptions 2-3 are standard in recent literature [11, 23, 26, 28, 29, 30, 32, 39, 41, 43]. For some specific examples, we refer the readers to [29, Section V] and [24, Section VI].

3. ADBB Development

3.1. The Barzilai-Borwein Step-Sizes

The original form of BB method solving problem (1) updates as follows:

$$\tilde{x}_{k+1} = \tilde{x}_k - \alpha_k \nabla f(\tilde{x}_k), \quad (6)$$

where α_k can be calculated by $\alpha_k = (\tilde{s}_k^\top \tilde{s}_k) / (\tilde{s}_k^\top \tilde{z}_k)$ or $\alpha_k = (\tilde{s}_k^\top \tilde{z}_k) / (\tilde{z}_k^\top \tilde{z}_k)$ in [36]. Therein $\tilde{s}_k = \tilde{x}_k - \tilde{x}_{k-1}$ and $\tilde{z}_k = \nabla f(\tilde{x}_k) - \nabla f(\tilde{x}_{k-1})$ for $k \geq 1$. The BB method has the superiority of simplicity and flexibility. In this paper, we set the distributed BB step-sizes as follows:

$$\alpha_k^i = \frac{1}{m} \frac{(s_k^i)^\top s_k^i}{(s_k^i)^\top v_k^i}, \quad (7)$$

$$\alpha_k^i = \frac{1}{m} \frac{(s_k^i)^\top v_k^i}{(v_k^i)^\top v_k^i}, \quad (8)$$

where $s_k^i = x_k^i - x_{k-1}^i$ and $v_k^i = \nabla f_i(x_k^i) - \nabla f_i(x_{k-1}^i)$. One may be aware that the denominators of Eqs. 7-8 are tending to zero, which in fact does not affect the bounds on α_k^i and we give the bounds in Lemma 1.

Remark 3. Compared with the BB step-sizes employed in DGM-BB-C [33], ADBB employs a relatively smaller step-size given in Lemma 1 ($1/m$ that of in DGM-BB-C [33]) to guarantee the convergence. In fact, the division of m is exactly caused by the unbalanced directed network. That is, when the unbalanced directed network degrades to undirected or balanced directed, the row-stochastic weight matrix reduces to doubly stochastic weight matrix and thus ADBB reduces to DGM-BB-C which attains larger step-sizes and a network-independent range of step-sizes.

3.2. Distributed Optimization Using Only Row-Stochastic Weight Matrices

Distributed optimization algorithms using row-stochastic matrices are based on the method proposed in [23], for instance, FROST [30] and [32] extends [23] by employing uncoordinated step-sizes. The distributed form of FROST [30] updates as follows:

$$x_{k+1}^i = \sum_{j=1}^m a_{ij} x_k^j - \alpha_i z_k^i, \quad (9a)$$

$$y_{k+1}^i = \sum_{j=1}^m a_{ij} y_k^j, \quad (9b)$$

$$z_{k+1}^i = \sum_{j=1}^m a_{ij} z_k^j + \frac{\nabla f_i(x_{k+1}^i)}{[y_{k+1}^i]_i} - \frac{\nabla f_i(x_k^i)}{[y_k^i]_i}, \quad (9c)$$

where each agent i maintains three variables: $x_k^i, z_k^i \in \mathbb{R}^n$ and $y_k^i \in \mathbb{R}^m$, and α_i is the uncoordinated step-size locally selected by agent i , $i \in \mathcal{V}$. Note that a_{ij} in (9) is the weight assigned by agent i to the information from agent j . The row-stochastic weights, $A^{ij} = \{a_{ij}\}$, are aligned with (2).

Remark 4. The methods [19, 26, 24] constructing only column-stochastic weight matrices $B = [b_{ij}] \in \mathbb{R}^{m \times m}$, where $b_{ij} = \begin{cases} > 0, & i \in \mathcal{N}_j^{\text{out}} \\ 0, & \text{otherwise} \end{cases}$, $\sum_{i=1}^m b_{ij} = 1, \forall j \in \mathcal{V}$, require all agents to know their out-degrees and thus are not practical to broadcast-based communication protocol. Therefore, distributed optimization using only row-stochastic weight matrices is the state-of-the-art method, which has significant contributions than doubly or column-stochastic distributed optimization

(see [23, 30, 43] for details). Inspired by [23, 30], this paper extends [33] to unbalanced directed networks.

3.3. ADBB for Distributed Optimization

Based on the above analysis, we develop ADBB for distributed optimization over unbalanced directed networks in Algorithm 1.

Remark 5. We emphasize that the initialization of auxiliary variables $y_0^i = e_i$, $i \in \mathcal{V}$ requires each agent in the system has a unique identifier which is satisfied under Assumption 1. The initial values of step-sizes α_0^i , $i \in \mathcal{V}$ should be positive and Section 5 numerically demonstrates that ADBB is not sensitive to the initial values of α_0^i .

4. Convergence Analysis

Algorithm 1 ADBB for each agent $i \in \mathcal{V}$

Initialization: For each agent $i \in \mathcal{V}$, $x_0^i \in \mathbb{R}^n$ is arbitrary; $y_0^i = e_i \in \mathbb{R}^m$; $z_0^i = \nabla f_i(x_0^i) \in \mathbb{R}^n$; $\alpha_0^i > 0$; Choose a_{ij} with $\sum_{j \in \mathcal{N}_i^{\text{in}}} a_{ij} = 1$, for $\forall i, j \in \mathcal{V}$; $h = 1, 2, \dots, H$.

For $k = 0, 1, \dots$ **do**

Each agent $i \in \mathcal{V}$ **computes:** $x_{k+1}^i(0) = x_k^i - \alpha_k^i z_k^i$

Each agent $i \in \mathcal{V}$ **communicates:** $x_{k+1}^i(h) = \sum_{j \in \mathcal{N}_i^{\text{in}} \cup \{i\}} a_{ij} x_{k+1}^j(h-1)$

where α_k^i is computed by (7) or (8), and set $x_{k+1}^i = x_{k+1}^i(H)$.

Each agent $i \in \mathcal{V}$ **computes:** $y_{k+1}^i(0) = y_k^i$

Each agent $i \in \mathcal{V}$ **communicates:** $y_{k+1}^i(h) = \sum_{j \in \mathcal{N}_i^{\text{in}} \cup \{i\}} a_{ij} y_{k+1}^j(h-1)$

setting $y_{k+1}^i = y_{k+1}^i(H)$,

Each agent $i \in \mathcal{V}$ **computes:** $z_{k+1}^i(0) = z_k^i + \frac{\nabla f_i(x_{k+1}^i)}{[y_{k+1}^i]_i} - \frac{\nabla f_i(x_k^i)}{[y_k^i]_i}$

Each agent $i \in \mathcal{V}$ **communicates:** $z_{k+1}^i(h) = \sum_{j \in \mathcal{N}_i^{\text{in}} \cup \{i\}} a_{ij} z_{k+1}^j(h-1)$

and set $z_{k+1}^i = z_{k+1}^i(H)$.

End

In this section, the convergence analysis of ADBB is presented. For simplifying the analysis, we first make definitions as follows:

$$x_k := \left[(x_k^1)^\top, (x_k^2)^\top, \dots, (x_k^m)^\top \right]^\top,$$

$$\tilde{y}_k := [y_k^1, y_k^2, \dots, y_k^m]^\top,$$

$$z_k := \left[(z_k^1)^\top, (z_k^2)^\top, \dots, (z_k^m)^\top \right]^\top,$$

$$y_k := \tilde{y}_k \otimes I_n,$$

$$\hat{y}_k := \text{diag} \{y_k\},$$

$$\alpha_k := [\alpha_k^1, \alpha_k^2, \dots, \alpha_k^m]^\top,$$

$$D_k^\alpha := \text{diag} \{\alpha_k\} \otimes I_n,$$

$$\mathcal{A} := A \otimes I_n \in \mathbb{R}^{mn \times mn},$$

$$\nabla F(x_k) := \left[\nabla f_1(x_k^1)^\top, \nabla f_2(x_k^2)^\top, \dots, \nabla f_m(x_k^m)^\top \right]^\top,$$

$$\nabla F(x^*) := \left[\nabla f_1(x^*)^\top, \nabla f_2(x^*)^\top, \dots, \nabla f_m(x^*)^\top \right]^\top,$$

where $x_k, \tilde{y}_k, z_k, \alpha_k, \nabla F(x_k), \nabla F(x^*)$, simultaneously collect their local variables. Then, based on the above definitions, we give the compact form of ADBB as follows:

$$x_{k+1} = \mathcal{A}^H (x_k - D_k^\alpha z_k), \quad (10a)$$

$$\tilde{y}_{k+1} = A^H \tilde{y}_k, \quad (10b)$$

$$z_{k+1} = \mathcal{A}^H (z_k + \hat{y}_{k+1}^{-1} \nabla F(x_{k+1}) - \hat{y}_k^{-1} \nabla F(x_k)), \quad (10c)$$

where $\tilde{y}_0 = I_m$, $z_0 = \nabla f(x_0)$ and x_0 is arbitrary. This paper extends one dimension in [23] to n dimensions. To proceed, we continue to define

$$y_\infty := \lim_{k \rightarrow \infty} y_k,$$

$$\hat{y}_\infty := \text{diag} \{y_\infty\},$$

$$Y := \sup_{k \geq 0} \|\hat{y}_k^{-1}\|_2,$$

$$\hat{Y} := \sup_{k \geq 0} \|y_k\|_2,$$

$$p_1 := \|I_{mn} - \mathcal{A}^H\|_2,$$

$$\alpha_{\max} := \max_{k \geq 0} \{\alpha_k^i\},$$

$$\bar{\alpha}_{\max} := \max_{k \geq 0} \left\{ \frac{1}{m} \sum_{i=1}^m \alpha_k^i \right\},$$

where y_∞, Y, \hat{Y} exist due to the primitive row-stochastic matrix A .

4.1. Auxiliary Relations

Lemma 1. *Supposing that Assumptions 2-3 hold, for $\forall k \geq 0$, the BB step-size $\alpha_k^i, i \in \mathcal{V}$ computed by (7) or (8) in ADBB satisfies*

$$\frac{1}{mL_f} \leq \alpha_k^i \leq \frac{1}{m\mu}. \quad (11)$$

Proof 1. *To begin with, we derive bounds on the BB step-size α_k^i following (7). According to the strong convexity of local objective function f_i , it holds that*

$$(x_k^i - x_{k-1}^i)^\top (\nabla f_i(x_k^i) - \nabla f_i(x_{k-1}^i)) \geq \mu \|x_k^i - x_{k-1}^i\|_2^2. \quad (12)$$

Then, the upper bound for each BB step-size α_k^i is derived as follows:

$$\begin{aligned} \alpha_k^i &= \frac{1}{m} \frac{(x_k^i - x_{k-1}^i)^\top (x_k^i - x_{k-1}^i)}{(x_k^i - x_{k-1}^i)^\top (\nabla f_i(x_k^i) - \nabla f_i(x_{k-1}^i))} \\ &\leq \frac{1}{m} \frac{\|x_k^i - x_{k-1}^i\|_2^2}{\mu \|x_k^i - x_{k-1}^i\|_2^2} \\ &= \frac{1}{m\mu}, \end{aligned} \quad (13)$$

and according to Lipschitz continuity and Cauchy inequality, the lower bound for each BB step-size α_k^i is derived as follows:

$$\begin{aligned} \alpha_k^i &= \frac{1}{m} \frac{(x_k^i - x_{k-1}^i)^\top (x_k^i - x_{k-1}^i)}{(x_k^i - x_{k-1}^i)^\top (\nabla f_i(x_k^i) - \nabla f_i(x_{k-1}^i))} \\ &\geq \frac{1}{m} \frac{\|x_k^i - x_{k-1}^i\|_2^2}{L_f \|x_k^i - x_{k-1}^i\|_2^2} \\ &= \frac{1}{mL_f}. \end{aligned} \quad (14)$$

Next, we derive the bounds on BB step-size α_k^i following (8). According to Lipschitz continuity, it holds that

$$L_f (x_k^i - x_{k-1}^i)^\top (\nabla f_i(x_k^i) - \nabla f_i(x_{k-1}^i)) \geq \|\nabla f_i(x_k^i) - \nabla f_i(x_{k-1}^i)\|_2^2. \quad (15)$$

Then, step-size α_k^i is lower bounded by $1/(mL_f)$. Clearly, the upper bound on step-size α_k^i is no greater than $1/(m\mu)$. The above proofs show that the range of step-size α_k^i computed by (7) is longer than the range of step-size α_k^i computed by (8). Recalling the definitions of α_{\max} and $\bar{\alpha}_{\max}$, we can directly obtain that

$$\frac{1}{mL_f} \leq \alpha_{\max}, \bar{\alpha}_{\max} \leq \frac{1}{m\mu}. \quad (16)$$

The proof is completed.

Remark 6. It is worth mentioning that owing to the availability of larger step-sizes, ADBB is competitive than the state-of-art algorithms [19, 23, 26, 28, 29, 30, 32, 41, 43] that can be applied into unbalanced directed networks. Especially, the largest step-sizes employed by [23, 24, 26, 30] do not exceed $1/(mL_f)$ which is exactly the lower bound of the step-size employed in ADBB.

Lemma 2. Supposing that Assumption 1 holds, recalling the definitions of augmented weight matrix $\mathcal{A} = A \otimes I_n$ and y_∞ , for $\forall x \in \mathbb{R}^{mn}$, there holds

$$\|\mathcal{A}^H x - y_\infty x\|_\pi \leq \sigma^H \|x - y_\infty x\|_\pi, \quad (17)$$

where $0 < \sigma := \|\mathcal{A} - y_\infty\|_\pi < 1$ serves as the mixing rate of the directed network.

Proof 2. Since A is primitive and row-stochastic, it holds that $y_\infty = (1_m \pi^\top) \otimes I_n$ from Perron-Frobenius theorem [30]. Then, in terms of (10b), we get that $\mathcal{A}^H y_\infty = y_\infty$ and $y_\infty y_\infty = y_\infty$, which yields that

$$\mathcal{A}^H x - y_\infty x = (\mathcal{A}^H - y_\infty) (x - y_\infty x). \quad (18)$$

We know that $\rho(\mathcal{A} - y_\infty) < 1$ and

$$\begin{aligned} \mathcal{A}^H - y_\infty &= (\mathcal{A}^{H-1} - y_\infty) (\mathcal{A} - y_\infty) \\ &= (\mathcal{A} - y_\infty)^H. \end{aligned} \quad (19)$$

Then, it is easy to obtain that $\|\mathcal{A}^H - y_\infty\|_\pi < 1$. The proof follows via setting $\sigma = \|\mathcal{A} - y_\infty\|_\pi$.

Lemma 3. *Suppose that Assumption 1 holds. Recalling the definitions of y_k and y_∞ , it holds*

$$\|y_k - y_\infty\|_2 \leq \sigma^{kH}, \forall k \geq 0, \quad (20)$$

where $0 < \sigma^H < 1$ is a constant.

Proof 3. *In terms of (10b), we obtain that $y_k = A^{kH} \otimes I_n = \mathcal{A}^{kH}$. Similar with the proof in Lemma 2, it holds that*

$$\|y_k - y_\infty\|_2 = \left\| (\mathcal{A}^H - y_\infty)^k \right\|_2 \leq \sigma^{kH}, \forall k \geq 0. \quad (21)$$

The proof is completed.

Lemma 4. *Suppose that Assumption 1 holds. For $\forall k \geq 0$, the following inequalities hold*

$$\|\hat{y}_k^{-1} - \hat{y}_\infty^{-1}\|_2 \leq \sqrt{m} \sigma^{kH} Y^2, \quad (22)$$

$$\|\hat{y}_{k+1}^{-1} - \hat{y}_k^{-1}\|_2 \leq 2\sqrt{m} \sigma^{kH} Y^2. \quad (23)$$

Proof 4. *We first give the proof of (22) as follows:*

$$\begin{aligned} \|\hat{y}_k^{-1} - \hat{y}_\infty^{-1}\|_2 &= \|\hat{y}_k^{-1} (\hat{y}_\infty - \hat{y}_k) \hat{y}_\infty^{-1}\|_2 \\ &\leq \|\hat{y}_k^{-1}\|_2 \|\text{diag}\{y_k - y_\infty\}\|_2 \|\hat{y}_\infty^{-1}\|_2 \\ &\leq \sqrt{m} \sigma^{kH} Y^2. \end{aligned} \quad (24)$$

The proof of (23) directly follows from the proof of (22), i.e.,

$$\begin{aligned} \|\hat{y}_{k+1}^{-1} - \hat{y}_k^{-1}\|_2 &= \|\hat{y}_{k+1}^{-1} - \hat{y}_\infty^{-1} + \hat{y}_\infty^{-1} - \hat{y}_k^{-1}\|_2 \\ &\leq \|\hat{y}_{k+1}^{-1} - \hat{y}_\infty^{-1}\|_2 + \|\hat{y}_\infty^{-1} - \hat{y}_k^{-1}\|_2 \\ &\leq 2\sqrt{m} \sigma^{kH} Y^2, \end{aligned} \quad (25)$$

where the first inequality utilizes the triangle inequality.

Lemma 5. *Suppose that Assumption 1 holds. For $\forall k \geq 0$, the following equation holds*

$$y_\infty z_k = y_\infty \hat{y}_k^{-1} \nabla F(x_k). \quad (26)$$

Proof 5. Recalling that $y_\infty \mathcal{A}^H = y_\infty$, from (10c), we recursively update

$$\begin{aligned}
y_\infty z_k &= y_\infty z_{k-1} + y_\infty \hat{y}_k^{-1} \nabla F(x_k) - y_\infty \hat{y}_{k-1}^{-1} \nabla F(x_{k-1}) \\
&= y_\infty z_{k-2} + y_\infty \hat{y}_{k-1}^{-1} \nabla F(x_{k-1}) - y_\infty \hat{y}_{k-2}^{-1} \nabla F(x_{k-2}) \\
&\quad + y_\infty \hat{y}_k^{-1} \nabla F(x_k) - y_\infty \hat{y}_{k-1}^{-1} \nabla F(x_{k-1}) \\
&= y_\infty \hat{y}_k^{-1} \nabla F(x_k),
\end{aligned} \tag{27}$$

where the last equality follows from the initial conditions that $\hat{y}_0 = I_{mn}$ and $z_0 = \nabla F(x_0)$.

Lemma 6. ([44]) Let the global objective function $f(x)$ be μ -strongly convex and L_f -Lipschitz continuous, where $0 < \mu \leq L_f$. Then, for $\forall x \in \mathbb{R}^m$ and $0 < \theta < \frac{2}{L_f}$, we have

$$\|x - \theta \nabla f(x) - x^*\|_2 \leq \eta \|x - x^*\|_2, \tag{28}$$

where $\eta = \max\{|1 - \theta\mu|, |1 - \theta L_f|\}$.

Lemma 7. Suppose that Assumption 1 holds. For $\forall k \geq 0$, the following inequality holds

$$\begin{aligned}
\|z_k\|_2 &\leq \bar{\pi}^{0.5} m L_f \|x_k - y_\infty x_k\|_\pi + m L_f \|y_\infty x_k - \mathbf{1}_m \otimes \tilde{x}^*\|_2 \\
&\quad + \underline{\pi}^{-0.5} \|z_k - y_\infty z_k\|_\pi + \sqrt{m} \hat{Y} Y^2 \sigma^{kH} \|\nabla F(x_k)\|_2.
\end{aligned} \tag{29}$$

Proof 6. Recalling $y_\infty \hat{y}_\infty^{-1} = (\mathbf{1}_m \mathbf{1}_m^\top) \otimes I_n$ and $y_\infty z_k = y_\infty \hat{y}_k^{-1} \nabla F(x_k)$ from Lemma 5, we obtain that

$$\begin{aligned}
\|z_k\|_2 &\leq \|z_k - y_\infty z_k\|_2 + \|y_\infty z_k\|_2 \\
&\leq \underline{\pi}^{-0.5} \|z_k - y_\infty z_k\|_\pi + \|y_\infty \hat{y}_k^{-1} \nabla F(x_k) - y_\infty \hat{y}_\infty^{-1} \nabla F(x_k)\|_2 \\
&\quad + \|y_\infty \hat{y}_\infty^{-1} \nabla F(x_k) - (\mathbf{1}_m \mathbf{1}_m^\top) \otimes I_n \nabla F(x^*)\|_2, \\
&\leq \underline{\pi}^{-0.5} \|z_k - y_\infty z_k\|_\pi + \|y_\infty \hat{y}_k^{-1} - y_\infty \hat{y}_\infty^{-1}\|_2 \|\nabla F(x_k)\|_2 \\
&\quad + m \|\nabla F(x_k) - \nabla F(x^*)\|_2,
\end{aligned}$$

where the second inequality utilizes the fact that $(\mathbf{1}_m^\top \otimes I_n) \nabla F(x^*) = 0$. Applying Lipschitz continuity and (22) completes the proof.

To proceed, the contraction relationship of ADBB is established by deriving upper bounds on the sequel quantities: i) the consensus error: $\|x_{k+1} - y_\infty x_{k+1}\|_\pi$; ii) the optimality gap: $\|y_\infty x_{k+1} - 1_m \otimes \tilde{x}^*\|_2$; iii) the gradient tracking error: $\|z_{k+1} - y_\infty z_{k+1}\|_\pi$.

4.2. Contraction Relationship

Lemma 8. *Suppose that Assumption 1 holds. For $\forall k \geq 0$, the following inequality holds*

$$\begin{aligned} & \|x_{k+1} - y_\infty x_{k+1}\|_\pi \\ & \leq (1 + \alpha_{\max} \bar{\pi} m L_f) \sigma^H \|x_k - y_\infty x_k\|_\pi + \alpha_{\max} \bar{\pi}^{0.5} m L_f \sigma^H \|y_\infty x_k - 1_m \otimes \tilde{x}^*\|_2 \\ & \quad + \alpha_{\max} \vartheta^{0.5} \sigma^H \|z_k - y_\infty z_k\|_\pi + \alpha_{\max} \sqrt{m \bar{\pi}} \hat{Y} Y^2 \sigma^{(k+1)H} \|\nabla F(x_k)\|_2. \end{aligned} \quad (30)$$

Proof 7. *According to (10a) and $y_\infty \mathcal{A}^H = y_\infty$, it holds that*

$$\begin{aligned} & \|x_{k+1} - y_\infty x_{k+1}\|_\pi \\ & = \|\mathcal{A}^H(x_k - D_k^\alpha z_k) - y_\infty(x_k - D_k^\alpha z_k)\|_\pi \\ & = \|(\mathcal{A}^H - y_\infty)(x_k - y_\infty x_k) + (\mathcal{A}^H - y_\infty) D_k^\alpha z_k\|_\pi \\ & \leq \sigma^H \|x_k - y_\infty x_k\|_\pi + \alpha_{\max} \bar{\pi}^{0.5} \sigma^H \|z_k\|_2, \end{aligned} \quad (31)$$

where the second equality follows the relationship $(\mathcal{A}^H - y_\infty) y_\infty = 0$ and the inequality applies Lemma 2. The proof is finished through substituting (29) into (31).

Lemma 9. *Suppose that Assumptions 1-3 hold. If $\pi^\top \alpha_k < 2/(m L_f)$, for $\forall k \geq 0$, then the following inequality holds*

$$\begin{aligned} & \|y_\infty x_{k+1} - 1_m \otimes \tilde{x}^*\|_2 \\ & \leq \alpha_{\max} \bar{\pi}^{-0.5} m L_f \|x_k - y_\infty x_k\|_\pi + \lambda \|y_\infty x_k - 1_m \otimes \tilde{x}^*\|_2 \\ & \quad + \alpha_{\max} \bar{\pi}^{-0.5} \hat{Y} \|z_k - y_\infty z_k\|_\pi + \alpha_{\max} \sqrt{m \bar{\pi}} \hat{Y} Y^2 \sigma^{kH} \|\nabla F(x_k)\|_2, \end{aligned} \quad (32)$$

where $\lambda = \max\{|1 - m \pi^\top \alpha_k \mu|, |1 - m \pi^\top \alpha_k L_f|\}$.

Proof 8. Recalling $y_\infty \mathcal{A}^H = y_\infty$, we obtain that

$$\begin{aligned}
& \|y_\infty x_{k+1} - 1_m \otimes \tilde{x}^*\|_2 \\
&= \|y_\infty (\mathcal{A}^H (x_k - D_k^\alpha z_k) + (D_k^\alpha - D_k^\alpha) y_\infty z_k) - 1_m \otimes \tilde{x}^*\|_2 \\
&\leq \|y_\infty x_k - y_\infty D_k^\alpha y_\infty z_k - 1_m \otimes \tilde{x}^*\|_2 + \alpha_{\max} \hat{Y} \|z_k - y_\infty z_k\|_2 \\
&\leq \|y_\infty x_k - y_\infty D_k^\alpha y_\infty z_k - 1_m \otimes \tilde{x}^*\|_2 + \alpha_{\max} \hat{Y} \underline{\pi}^{-0.5} \|z_k - y_\infty z_k\|_\pi.
\end{aligned} \tag{33}$$

We know that

$$\begin{aligned}
y_\infty D_k^\alpha y_\infty &= ((1_m \pi^\top) \otimes I_n) (\text{diag} \{\alpha_k\} \otimes I_n) ((1_m \pi^\top) \otimes I_n) \\
&= (\pi^\top \alpha_k) y_\infty.
\end{aligned} \tag{34}$$

We continue to bound $\|y_\infty x_k - y_\infty D_k^\alpha y_\infty z_k - 1_m \otimes \tilde{x}^*\|_2$ as follows:

$$\begin{aligned}
& \|y_\infty x_k - y_\infty D_k^\alpha y_\infty z_k - 1_m \otimes \tilde{x}^*\|_2 \\
&\leq \sqrt{m} \|(\pi^\top \otimes I_n) x_k - \tilde{x}^* - m (\pi^\top \alpha_k) \nabla f((\pi^\top \otimes I_n) x_k)\|_2 \\
&\quad + \|m (\pi^\top \alpha_k) (1_m \otimes I_n) \nabla f((\pi^\top \otimes I_n) x_k) - (\pi^\top \alpha_k) y_\infty z_k\|_2 \\
&= \Delta_1 + \Delta_2.
\end{aligned} \tag{35}$$

If $\pi^\top \alpha_k < 2/(mL_f)$, then using Lemma 6, it holds that

$$\Delta_1 \leq \lambda \|y_\infty x_k - 1_m \otimes \tilde{x}^*\|_2, \tag{36}$$

where $\lambda = \max \{|1 - m\pi^\top \alpha_k \mu|, |1 - m\pi^\top \alpha_k L_f|\}$. Next, we aim to bound Δ_2 .

$$\begin{aligned}
\Delta_2 &= (\pi^\top \alpha_k) \|m(1_m \otimes I_n) \nabla f((\pi^\top \otimes I_n) x_k) - y_\infty z_k\|_2 \\
&\leq \alpha_{\max} \|m(1_m \otimes I_n) \nabla f((\pi^\top \otimes I_n) x_k) - (1_m \otimes I_n) (1_m^\top \otimes I_n) \nabla F(x_k)\|_2 \\
&\quad + \alpha_{\max} \|(1_m \otimes I_n) (1_m^\top \otimes I_n) \nabla F(x_k) - y_\infty \hat{y}_k^{-1} \nabla F(x_k)\|_2 \\
&\leq \alpha_{\max} mL_f \|x_k - y_\infty x_k\|_2 + \alpha_{\max} \sqrt{m} \hat{Y} Y^2 \sigma^{kH} \|\nabla F(x_k)\|_2 \\
&\leq \alpha_{\max} \underline{\pi}^{-0.5} mL_f \|x_k - y_\infty x_k\|_\pi + \alpha_{\max} \sqrt{m} \hat{Y} Y^2 \sigma^{kH} \|\nabla F(x_k)\|_2,
\end{aligned} \tag{37}$$

where the first inequality uses Lemma 5 and the last inequality applies Lipschitz continuity and (22). Plugging (36) and (37) into (35) yields

$$\begin{aligned}
& \|y_\infty x_k - y_\infty D_k^\alpha y_\infty z_k - 1_m \otimes \tilde{x}^*\|_2 \\
&\leq \alpha_{\max} \underline{\pi}^{-0.5} mL_f \|x_k - y_\infty x_k\|_\pi + \lambda \|y_\infty x_k - 1_m \otimes \tilde{x}^*\|_2 \\
&\quad + \alpha_{\max} \sqrt{m} \hat{Y} Y^2 \sigma^{kH} \|\nabla F(x_k)\|_2.
\end{aligned} \tag{38}$$

The proof is completed by substituting (38) into (33).

Lemma 10. *Suppose that Assumptions 1-2 hold. For $\forall k \geq 0$, the following inequality holds*

$$\begin{aligned}
\|z_{k+1} - y_\infty z_{k+1}\|_\pi &\leq \left(\vartheta^{0.5} Y L_f p_1 + \alpha_{\max} \bar{\pi}^{0.5} m Y \hat{Y} L_f^2 \right) \sigma^H \|x_k - y_\infty x_k\|_\pi \\
&\quad + \alpha_{\max} m Y \hat{Y} L_f^2 \sigma^H \|y_\infty x_k - \mathbf{1}_m \otimes \tilde{x}^*\|_2 \\
&\quad + \left(1 + \alpha_{\max} \underline{\pi}^{-0.5} Y \hat{Y} L_f \right) \sigma^H \|z_k - y_\infty z_k\|_\pi \\
&\quad + \left(2\sqrt{m\underline{\pi}^{-1}} + \alpha_{\max} \sqrt{m} L_f Y \hat{Y}^2 \right) Y^2 \sigma^{(k+1)H} \|\nabla F(x_k)\|_2
\end{aligned} \tag{39}$$

Proof 9. *According to Lemma 2 and (10c), we obtain that*

$$\begin{aligned}
&\|z_{k+1} - y_\infty z_{k+1}\|_\pi \\
&= \|\mathcal{A}^H (z_k + \hat{y}_{k+1}^{-1} \nabla F(x_{k+1}) - \hat{y}_k^{-1} \nabla F(x_k)) - y_\infty z_{k+1}\|_\pi \\
&\leq \|\mathcal{A}^H (y_{k+1}^{-1} \nabla F(x_{k+1}) - \hat{y}_k^{-1} \nabla F(x_k)) - (y_\infty z_{k+1} - y_\infty z_k)\|_\pi \\
&\quad + \sigma^H \|z_k - y_\infty z_k\|_\pi.
\end{aligned} \tag{40}$$

Recall that $y_\infty z_k = y_\infty \hat{y}_k^{-1} \nabla F(x_k)$ from Lemma 5. Therefore

$$\begin{aligned}
&\|\mathcal{A}^H (y_{k+1}^{-1} \nabla F(x_{k+1}) - \hat{y}_k^{-1} \nabla F(x_k)) - (y_\infty z_{k+1} - y_\infty z_k)\|_\pi \\
&= \|(\mathcal{A}^H - y_\infty) (y_{k+1}^{-1} \nabla F(x_{k+1}) - \hat{y}_k^{-1} \nabla F(x_k))\|_\pi \\
&\leq \sigma^H \|y_{k+1}^{-1} \nabla F(x_{k+1}) - y_{k+1}^{-1} \nabla F(x_k)\|_\pi + \sigma^H \|y_{k+1}^{-1} \nabla F(x_k) - \hat{y}_k^{-1} \nabla F(x_k)\|_\pi \\
&\leq \underline{\pi}^{-0.5} Y L_f \sigma^H \|x_{k+1} - x_k\|_2 + 2\underline{\pi}^{-0.5} \sqrt{m} Y^2 \sigma^{(k+1)H} \|\nabla F(x_k)\|_2.
\end{aligned} \tag{41}$$

We next bound $\|x_{k+1} - x_k\|$,

$$\begin{aligned}
&\|x_{k+1} - x_k\|_2 \\
&= \|\mathcal{A}^H (x_k - D_k^\alpha z_k) - x_k\|_2 \\
&\leq \|(I_{mn} - \mathcal{A}^H) (x_k - y_\infty x_k)\|_2 + \alpha_{\max} \|\mathcal{A}^H\|_2 \|z_k\|_2 \\
&\leq \bar{\pi}^{0.5} p_1 \|x_k - y_\infty x_k\|_\pi + \alpha_{\max} \hat{Y} \|z_k\|_2,
\end{aligned} \tag{42}$$

where the first inequality utilizes the fact that $(I_{mn} - \mathcal{A}^H) y_\infty = 0$. Combining

(41) and (42) gives

$$\begin{aligned} & \left\| \mathcal{A}^H (y_{k+1}^{-1} \nabla F(x_{k+1}) - \hat{y}_k^{-1} \nabla F(x_k)) - (y_\infty z_{k+1} - y_\infty z_k) \right\|_\pi \\ & \leq \vartheta^{0.5} p_1 Y L_f \sigma^H \|x_k - y_\infty x_k\|_\pi + 2\sqrt{m\pi^{-1}} Y^2 \sigma^{(k+1)H} \|\nabla F(x_k)\|_2 + \alpha_{\max} \hat{Y} \|z_k\|_2. \end{aligned} \quad (43)$$

Substituting (29) into (40) gives

$$\begin{aligned} & \|z_{k+1} - y_\infty z_{k+1}\|_\pi \\ & \leq \vartheta^{0.5} p_1 Y L_f \sigma^H \|x_k - y_\infty x_k\|_\pi + \sigma^H \|z_k - y_\infty z_k\|_\pi \\ & \quad + 2\sqrt{m\pi^{-1}} Y^2 \sigma^{(k+1)H} \|\nabla F(x_k)\|_2 + \alpha_{\max} \hat{Y} \|z_k\|_2. \end{aligned} \quad (44)$$

The proof is ended via substituting (29) into (44).

Lemma 11. ([29, Lemma 4]) *Let $X \in \mathbb{R}^{n \times n}$ be a non-negative matrix and $x \in \mathbb{R}^n$ be a positive vector. If $Xx < \omega x$, then $\rho(X) < \omega$.*

Before presenting the main results, we make some definitions for subsequent proofs as follows:

$$\begin{aligned} t_k &= \begin{bmatrix} \|x_k - y_\infty x_k\|_\pi \\ \|y_\infty x_k - 1_m \otimes \tilde{x}^*\|_2 \\ \|z_k - y_\infty z_k\|_\pi \end{bmatrix}, \quad g_k = \begin{bmatrix} \|\nabla F(x_k)\|_2 \\ 0 \\ 0 \end{bmatrix}, \\ M_k &= \begin{bmatrix} \sigma^H + \alpha_{\max} \sigma^H w_1 & \alpha_{\max} \sigma^H w_1 & \alpha_{\max} \sigma^H \\ \alpha_{\max} w_2 & \lambda & \alpha_{\max} w_3 \\ \sigma^H w_4 + \alpha_{\max} \sigma^H w_5 & \alpha_{\max} \sigma^H w_5 & \sigma^H + \alpha_{\max} \sigma^H w_6 \end{bmatrix}, \\ G_k &= \begin{bmatrix} \alpha_{\max} \sqrt{m\pi} \hat{Y} Y^2 \sigma^H & 0 & 0 \\ \alpha_{\max} \sqrt{m} Y^2 \hat{Y} & 0 & 0 \\ 2\sqrt{m\pi^{-1}} Y^2 \sigma^H + \alpha_{\max} \sqrt{m} \hat{Y}^2 Y^3 L_f \sigma^H & 0 & 0 \end{bmatrix} \sigma^{kH}, \end{aligned}$$

where $w_1 = \bar{\pi}^{0.5} m L_f$, $w_2 = \underline{\pi}^{-0.5} m L_f$, $w_3 = \underline{\pi}^{-0.5} \hat{Y}$, $w_4 = \vartheta^{0.5} Y L_f p_1$, $w_5 = m Y \hat{Y} L_f^2$, $w_6 = \underline{\pi}^{-0.5} Y \hat{Y} L_f$. Note that G_k decays linearly over k , since $0 < \sigma < 1$.

4.3. Main Results

Theorem 1. *Suppose that Assumptions 1-3 hold. If $\pi^\top \alpha_k < 2/(mL_f)$ for $\forall k \geq 0$, one can establish a linear time invariant inequality as follows:*

$$t_{k+1} \leq M_k t_k + G_k g_k, \forall k \geq 0, \quad (45)$$

where $t_k, g_k \in \mathbb{R}^3$, and $M_k, G_k \in \mathbb{R}^{3 \times 3}$ are defined before Theorem 1. Furthermore, when the largest step-size α_{\max} satisfies (16) with proper setting of the multi-consensus inner loop number H , there holds that $\rho(M_k) < 1$.

Proof 10. *Recall that $\lambda = \max \{|1 - m\pi^\top \alpha_k \mu|, |1 - m\pi^\top \alpha_k L_f|\}$. Clearly, we can obtain that $1/mL_f \leq \alpha_{\min} \leq \pi^\top \alpha_k$ from Lemma 1. If $\pi^\top \alpha_k \leq (2/mL_f) - (\mu/mL_f^2)$, it holds that $\lambda \leq 1 - (\mu/mL_f)$. Hence, we get that $M_k \leq M$, with*

$$M = \begin{bmatrix} \sigma^H + \alpha_{\max} \sigma^H w_1 & \alpha_{\max} \sigma^H w_1 & \alpha_{\max} \sigma^H \\ \alpha_{\max} w_2 & 1 - w_7 & \alpha_{\max} w_3 \\ \sigma^H w_4 + \alpha_{\max} \sigma^H w_5 & \alpha_{\max} \sigma^H w_5 & \sigma^H + \alpha_{\max} \sigma^H w_6 \end{bmatrix},$$

where $w_7 = \mu/L_f$. Thus, $\rho(M_k) \leq \rho(M)$. To proceed, we derive the lower bound on multi-consensus inner loop number H and solve for a positive vector $c = [c_1, c_2, c_3]^\top$ from

$$M \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} < \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, \quad (46)$$

which is equivalent to the following inequalities

$$\begin{cases} 0 < \alpha_{\max} (\sigma^H w_1 c_1 + \sigma^H w_1 c_2 + \sigma^H c_3) < (1 - \sigma^H) c_1, \\ 0 < \alpha_{\max} (w_2 c_1 + w_3 c_3) < w_7 c_2, \\ 0 < \alpha_{\max} (\sigma^H w_5 c_1 + \sigma^H w_5 c_2 + \sigma^H w_6 c_3) < (1 - \sigma^H) c_3 - \sigma^H w_4 c_1. \end{cases} \quad (47)$$

Since the right hand side of the last inequality in (47) has to be positive, it should hold that

$$0 < c_1 < \frac{(1 - \sigma^H) c_3}{\sigma^H w_4}. \quad (48)$$

According to (47), we can obtain the upper bound on the largest step-size as follows:

$$0 < \alpha_{\max} < \min \left\{ \frac{(1-\sigma^H) c_1}{\sigma^H (w_1 c_1 + w_1 c_2 + c_3)}, \frac{w_7 c_2}{w_2 c_1 + w_3 c_3}, \frac{(1-\sigma^H) c_3 - \sigma^H w_4 c_1}{\sigma^H (w_5 c_1 + w_5 c_2 + w_6 c_3)} \right\}, \quad (49)$$

where c_2, c_3 are arbitrary positive constants and c_1 is chosen from (48). According to (16), we need

$$\frac{1}{mL_f} \leq \alpha_{\max} \leq \frac{1}{m\mu}. \quad (50)$$

So as to guarantee that the range of largest step-size in (49) contains the range of the largest step-size in (50), the following inequality should hold

$$0 < \frac{1}{m\mu} < \min \left\{ \frac{(1-\sigma^H) c_1}{\sigma^H (w_1 c_1 + w_1 c_2 + c_3)}, \frac{w_7 c_2}{w_2 c_1 + w_3 c_3}, \frac{(1-\sigma^H) c_3 - \sigma^H w_4 c_1}{\sigma^H (w_5 c_1 + w_5 c_2 + w_6 c_3)} \right\}, \quad (51)$$

which yields

$$0 < \sigma^H < \min \left\{ \frac{m\mu c_1}{(w_1 + m\mu) c_1 + w_1 c_2 + c_3}, \frac{m\mu c_3}{(w_5 + m\mu w_4) c_1 + w_5 c_2 + (w_6 + m\mu) c_3} \right\}, \quad (52)$$

We define $\varpi = \min \left\{ \frac{m\mu c_1}{(w_1 + m\mu) c_1 + w_1 c_2 + c_3}, \frac{m\mu c_3}{(w_5 + m\mu w_4) c_1 + w_5 c_2 + (w_6 + m\mu) c_3} \right\}$. Then, it follows from (52) that

$$H \geq \left\lceil \frac{\ln \varpi}{\ln \sigma} \right\rceil + \varphi \left(\frac{\ln \varpi}{\ln \sigma} \right), \quad (53)$$

where $\varphi(x) = \begin{cases} 1, & \text{for } x \in \mathbb{N}_+ \\ 0, & \text{for } x \notin \mathbb{N}_+ \end{cases}$ and \mathbb{N}_+ denotes the set of positive integers. Thus,

it holds that $\rho(M) < 1$. Finally, we get $\rho(M_k) < 1$ due to $\rho(M_k) \leq \rho(M)$.

The proof is completed.

Remark 7. We emphasize that the BB step-size (see, (7) and (8)) is automatically computed by each agent in the system and only depends on the local information, which does not rely on the network parameter, σ . We also acknowledge that multi-consensus inner loop number H is lower bounded by the network parameter, σ , which cannot be computed locally. Therefore, we pick a sufficiently large multi-consensus number, H , to simultaneously guarantee the convergence

of ADBB and ensure that ADBB can run in a fully distributed manner. This notion of sufficiently large only serves for the conservative theoretical analysis while in practice we need to manually optimize the multi-consensus number, H , which is shown in Section 5.

Lemma 12. ([23, 30, Theorem 2]) Suppose that Assumptions 1-3 hold. Recall that $\rho(M_k) < 1$ according to Theorem 1 and G_k decays linearly over k since $0 < \sigma < 1$. Then, the sequence $\{x_k\}_{k \geq 0}$ generated by ADBB converges linearly to $x^* = 1_m \otimes \tilde{x}^*$, i.e., for some positive constant $\omega > 0$, it holds that

$$\|x_k - 1_m \otimes \tilde{x}^*\| \leq \omega (\max\{\rho(M_k), \sigma^H\} + \xi)^k, \forall k \geq 0, \quad (54)$$

where ξ is an arbitrary small constant such that $0 < \xi + \max\{\rho(M_k), \sigma^H\} < 1$.

5. Numerical Experiments

In order to confirm correctness of the theoretical analysis and show the performance of ADBB. We leverage regularized logistic regression to compare ADBB with some other well-known distributed optimization algorithms through solving a binary classification problem. Especially, a network of m agents cooperatively resolve a distributed logistic regression problem as follows:

$$\min_{\tilde{w} \in \mathbb{R}^n} f(\tilde{w}) = \frac{1}{m} \sum_{i=1}^m f_i(\tilde{w}), \quad (55)$$

where $\tilde{w} \in \mathbb{R}^n$ is the optimization variable to learn the separating hyperplane and each local objective function $f_i(\tilde{w})$ is expressed as

$$f_i(\tilde{w}) = \frac{1}{q_i} \sum_{j=1}^{q_i} \log(1 + \exp(-b_{ij} c_{ij}^\top \tilde{w})) + \frac{\beta}{2m} \|\tilde{w}\|_2^2, \quad (56)$$

where q_i is the number of local samples distributed to agent i ; $c_{ij} \in \mathbb{R}^n$ is the j -th training sample and $b_{ij} \in \{+1, -1\}$ is the corresponding label, both of which are only accessed by agent i ; β is the regularized constant. Therefore, the optimal solution to (55) is represented by

$$\tilde{w}^* = \arg \min_{\tilde{w} \in \mathbb{R}^n} \left(\frac{\beta}{2} \|\tilde{w}\|_2^2 + \frac{1}{m} \sum_{i=1}^m \frac{1}{q_i} \sum_{j=1}^{q_i} \log(1 + \exp(-b_{ij} c_{ij}^\top \tilde{w})) \right). \quad (57)$$

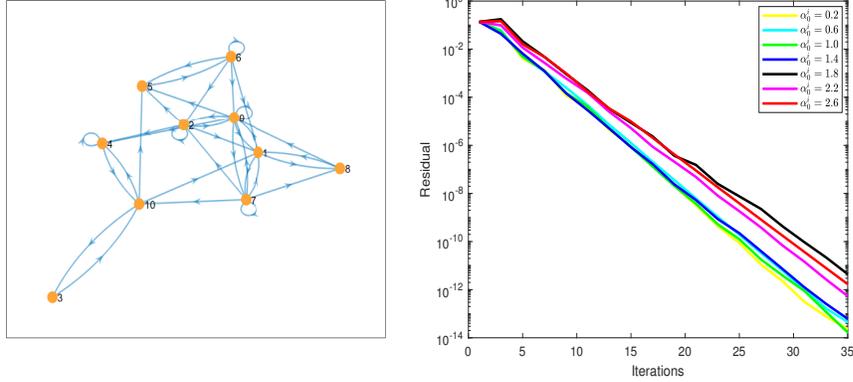


Fig. 1. An unbalanced directed network with **Fig. 2.** Performance of ADBB under different 10 agents. initial step-sizes α_0^i .

In the following experiments, we assume that the samples are distributed equally among the agents, i.e., $q_i = N/m$, $i \in \mathcal{V}$, where N is the total number of data and m represents the number of agents in the network. Then, we provide two case studies, in which the residual is defined as $(1/m) \sum_{i=1}^m \|w_k^i - \tilde{w}^*\|_2$. All simulations are carried out in MATLAB on a Lenovo laptop with 4.10 GHz, 4 Cores 8 Threads Intel i5 - 9300HF processor and 16GB memory.

Case Study 1

In the first case, the effect of the multi-consensus inner loop number H and initial step-sizes α_0^i , on ADBB is explored. Let $\mathcal{N}(\theta, \xi)$ to denote a normal distribution with the mean vector θ and the covariance matrix ξ . The total samples are $N = 1000$, and we set $m = 10$, $n = 100$. For each agent i , a half of sample vectors c_{ij} are generated by independent and identically distributed $\mathcal{N}([2, -2]^\top, 2I_n)$ with label $b_{ij} = +1$, while the others are sample vectors c_{ij} generated by independent and identically distributed $\mathcal{N}([-2, 2]^\top, 2I_n)$ with label $b_{ij} = -1$. Fig. 1 shows the unbalanced directed communication network; Fig. 2 compares the performance of ADBB with different initial step-sizes α_0^i by plotting the residual, where the multi-consensus inner loop number $H = 5$; Fig. 3 compares the performance of ADBB with different multi-consensus inner

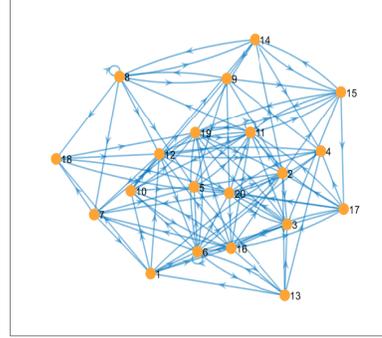
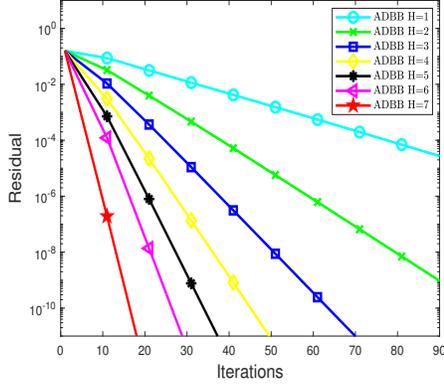


Fig. 3. Performance of ADBB under different multi-consensus inner loop numbers H .

Fig. 4. An unbalanced directed network with 20 agents.

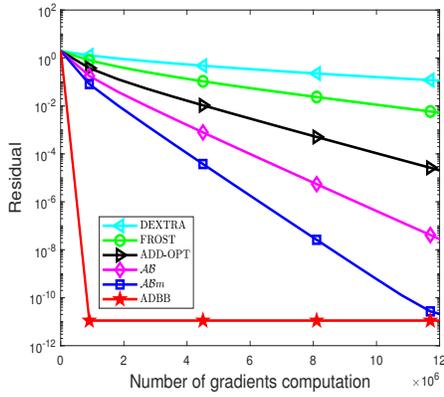


Fig. 5. Performance comparison over gradients computation.

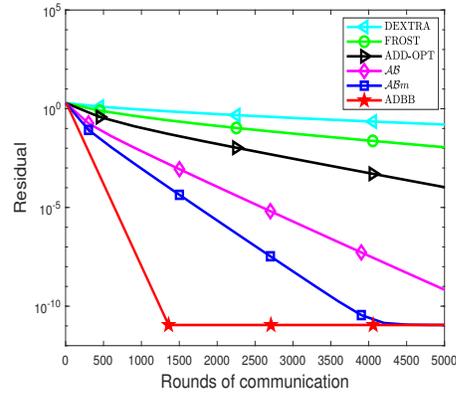


Fig. 6. Performance comparison over rounds of communication.

loop number H by plotting the residual, where the initial step-size $\alpha_0^i = 1.2$. The results in Figs. 2-3 show that ADBB is not sensitive to initial step-sizes α_0^i and ADBB converges faster when the multi-consensus inner loop number H increases, respectively.

Case Study 2

In the second case, ADBB and some existing algorithms are used to identify whether a mushroom is poisonous or not according to its different features, such

as cap-shape, cap-surface, cap-color, bruises, and so on. This case study is based on the mushroom data set provided in UCI Machine Learning Repository [45]. We randomly choose 8124 samples from the data set, from which $N = 6000$

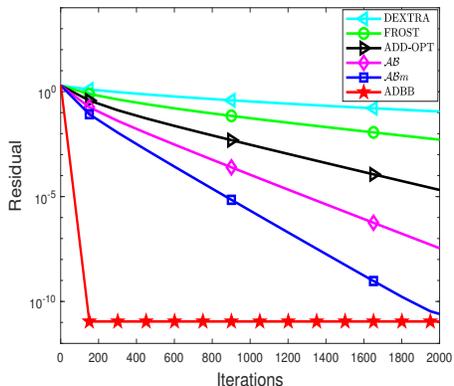


Fig. 7. Performance comparison over iterations.

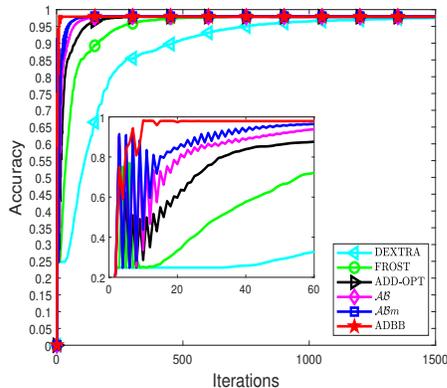


Fig. 8. Testing accuracy rate.

samples are used to train the discriminator and the rest of samples are used for testing. Each sample has $n = 112$ dimensions, which represents different features of the sample. We conduct the simulation in an unbalanced directed communication network as depicted in Fig. 4, where $m = 20$; initial step-size $\alpha_0^i = 1.2$, $i \in \mathcal{V}$; multi-consensus inner loop number $H = 3$. Assume label $b_{ij} = +1$ when the sample c_{ij} is poisonous; Assume label $b_{ij} = -1$ when the sample c_{ij} is edible. Figs. 5-7 demonstrate that ADBB is competitive than most existing work over unbalanced directed networks in terms of gradients computation, rounds of communication, and the number of iterations. The testing accuracy of different algorithms is plotted in Fig. 8. Tables 2-3 provide the confusion matrices to further clarify the testing results at iteration $k = 100$ and $k = 1000$, respectively. Notice that the total number of the testing samples is 2124 consisting of 1596 poisonous samples and 528 edible samples.

Predictive values \ True values	ADBB		$\mathcal{AB}m$		\mathcal{AB}		ADD-OPT		FROST		DEXTRA	
	P.	E.	P.	E.	P.	E.	P.	E.	P.	E.	P.	E.
P.	1587	9	1579	17	1563	33	1501	95	1297	299	553	1043
E.	36	492	34	494	34	494	36	492	12	516	0	528

P. is the abbreviation of poisonous and E. is the abbreviation of edible

Table 2: The confusion matrix (Testing results at iteration $k = 100$).

Predictive values \ True values	ADBB		$\mathcal{AB}m$		\mathcal{AB}		ADD-OPT		FROST		DEXTRA	
	P.	E.	P.	E.	P.	E.	P.	E.	P.	E.	P.	E.
P.	1589	7	1585	11	1579	17	1570	26	1564	32	1554	42
E.	28	500	30	498	30	498	32	496	32	496	30	498
The best accuracy among 20 experiments	0.98352		0.98069		0.97787		0.97928		0.97269		0.96610	
The average accuracy among 20 experiments	0.98021		0.97712		0.97710		0.97628		0.97124		0.96573	
The worse accuracy among 20 experiments	0.97012		0.96922		0.96975		0.96828		0.96823		0.96521	

P. is the abbreviation of poisonous and E. is the abbreviation of edible

Table 3: The confusion matrix (Testing results at iteration $k = 1000$).

6. Conclusions and future work

In this paper, a novel accelerated distributed algorithm constructing only row-stochastic weight matrices and using BB step-sizes, termed as ADBB, is developed to solve distributed convex optimization problems over an unbalanced directed network. Owing to the use of BB step-sizes and multi-consensus inner loops, ADBB allows each agent to automatically compute their step-sizes according to its local information and ensures the selection of larger step-sizes when achieving the globally optimal solution. Besides, ADBB has the fewer computation and communication costs than most existing distributed algorithms over unbalanced directed networks in simulations. To our knowledge, some existing variance-reduced stochastic gradient methods can protect the deterministic gradient algorithms from evaluating the full local gradients at each iteration and thus further reduce the computational costs. Therefore, as future work, it would be of considerable interest to incorporate the variance-reduced method into the proposed algorithm for distributed stochastic optimization over unbalanced directed networks.

Acknowledgments

The work described in this paper was supported in part by the Fundamental Research Funds for the Central Universities under Grant XDJK2019AC001, in part by the Innovation Support Program for Chongqing Overseas Returnees under Grant cx2019005, and in part by the National Natural Science Foundation of China under Grant 61773321, Grant 61673080.

References

- [1] H. Sun, M. Hong, Distributed non-convex first-order optimization and information processing: Lower complexity bounds and rate optimal algorithms, *IEEE Transactions on Signal Processing* 67 (22) (2019) 5912–5928.
- [2] V. Cevher, S. Becker, M. Schmidt, Convex optimization for big data: Scalable, randomized, and parallel algorithms for big data analytics, *IEEE Signal Processing Magazine* 31 (5) (2014) 32–43.
- [3] C. Tang, J. Ji, Y. Tang, S. Gao, Z. Tang, Y. Todo, A novel machine learning technique for computer-aided diagnosis, *Engineering Applications of Artificial Intelligence* 92 (2020) 103627.
- [4] D. L. Leottau, K. Lobos-Tsunekawa, F. Jaramillo, J. Ruiz-del Solar, Accelerating decentralized reinforcement learning of complex individual behaviors, *Engineering Applications of Artificial Intelligence* 85 (2019) 243–253.
- [5] S. Xie, Y. Xie, H. Ying, Z. Jiang, W. Gui, Neurofuzzy-based plant-wide hierarchical coordinating optimization and control: An application to zinc hydrometallurgy plant, *IEEE Transactions on Industrial Electronics* 67 (3) (2020) 2207–2219.
- [6] M. Ghasemi, I. F. Davoudkhani, E. Akbari, A. Rahimnejad, S. Ghavidel, L. Li, A novel and effective optimization algorithm for global optimization

and its engineering applications: Turbulent Flow of Water-based Optimization (TFWO), *Engineering Applications of Artificial Intelligence* 92 (2020) 103666.

- [7] H. Li, Z. Wang, G. Chen, Z. Y. Dong, Distributed Robust Algorithm for Economic Dispatch in Smart Grids over General Unbalanced Directed Networks, *IEEE Transactions on Industrial Informatics* 16 (7) (2019) 4322–4332.
- [8] H. Lee, S. H. Lee, T. Q. Quek, Deep learning for distributed optimization: Applications to wireless resource management, *IEEE Journal on Selected Areas in Communications* 37 (10) (2019) 2251–2266.
- [9] M. Blot, D. Picard, N. Thome, M. Cord, Distributed optimization for deep learning with gossip exchange, *Neurocomputing* 330 (2019) 287–296.
- [10] M. Liu, F. R. Yu, Y. Teng, V. C. Leung, M. Song, Performance optimization for blockchain-enabled industrial internet of things (iiot) systems: A deep reinforcement learning approach, *IEEE Transactions on Industrial Informatics* 15 (6) (2019) 3559–3570.
- [11] C. Xi, U. A. Khan, Distributed subgradient projection algorithm over directed graphs, *IEEE Transactions on Automatic Control* 62 (8) (2017) 3986–3992.
- [12] D. Yuan, D. W. Ho, G. P. Jiang, An adaptive primal-dual subgradient algorithm for online distributed constrained optimization, *IEEE Transactions on Cybernetics* 48 (11) (2017) 3045–3055.
- [13] N. Chatzipanagiotis, D. Dentcheva, M. M. Zavlanos, An augmented Lagrangian method for distributed optimization, *Mathematical Programming* 152 (1-2) (2015) 405–434.
- [14] A. Mokhtari, Q. Ling, A. Ribeiro, Network Newton distributed optimization methods, *IEEE Transactions on Signal Processing* 65 (1) (2016) 146–161.

- [15] A. Falsone, K. Margellos, S. Garatti, M. Prandini, Dual decomposition for multi-agent distributed optimization with coupling constraints, *Automatica* 84 (2017) 149–158.
- [16] F. Iutzeler, P. Bianchi, P. Ciblat, W. Hachem, Explicit convergence rate of a distributed alternating direction method of multipliers, *IEEE Transactions on Automatic Control* 61 (4) (2016) 892–904.
- [17] Q. Ling, W. Shi, G. Wu, A. Ribeiro, DLM: Decentralized linearized alternating direction method of multipliers, *IEEE Transactions on Signal Processing* 63 (15) (2015) 4051–4064.
- [18] W. Shi, Q. Ling, K. Yuan, G. Wu, W. Yin, On the linear convergence of the ADMM in decentralized consensus optimization, *IEEE Transactions on Signal Processing* 62 (7) (2014) 1750–1761.
- [19] A. Nedić, A. Olshevsky, Shi, Wei, Achieving geometric convergence for distributed optimization over time-varying graphs, *SIAM Journal on Optimization* 27 (4) (2017) 2597–2633.
- [20] W. Shi, Q. Ling, G. Wu, W. Yin, EXTRA: An exact first-order algorithm for decentralized consensus optimization, *SIAM Journal on Optimization* 25 (2) (2015) 944–966.
- [21] D. Jakovetić, J. M. F. Moura, J. Xavier, Linear convergence rate of a class of distributed augmented lagrangian algorithms, *IEEE Transactions on Automatic Control* 60 (4) (2014) 922–936.
- [22] J. Xu, S. Zhu, Y. C. Soh, L. Xie, Augmented distributed gradient methods for multi-agent optimization under uncoordinated constant stepsizes, in: 2015 54th IEEE Conference on Decision and Control (CDC), IEEE, 2015, pp. 2055–2060.
- [23] C. Xi, V. S. Mai, R. Xin, E. H. Abed, U. A. Khan, Linear convergence in optimization over directed graphs with row-stochastic matrices, *IEEE Transactions on Automatic Control* 63 (10) (2018) 3558–3565.

- [24] C. Xi, Q. Wu, U. A. Khan, On the distributed optimization over directed networks, *Neurocomputing* 267 (26) (2017) 508–515.
- [25] A. Nedić, A. Olshevsky, Distributed optimization over time-varying directed graphs, *IEEE Transactions on Automatic Control* 60 (3) (2015) 601–615.
- [26] C. Xi, R. Xin, U. A. Khan, ADD-OPT: Accelerated distributed directed optimization, *IEEE Transactions on Automatic Control* 63 (5) (2018) 1329–1339.
- [27] C. Xi, U. A. Khan, DEXTRA: A fast algorithm for optimization over directed graphs, *IEEE Transactions on Automatic Control* 62 (10) (2017) 4980–4993.
- [28] R. Xin, U. A. Khan, A linear algorithm for optimization over directed graphs with geometric convergence, *IEEE Control Systems Letters* 2 (3) (2018) 315–320.
- [29] R. Xin, S. Member, U. A. Khan, S. Member, Distributed heavy-ball: A generalization and acceleration of first-order methods with gradient tracking, *IEEE Transactions on Automatic Control* 65 (6) (2019) 2627–2633.
- [30] R. Xin, C. Xi, U. A. Khan, FROST—Fast row-stochastic optimization with uncoordinated step-sizes, *EURASIP Journal on Advances in Signal Processing* (1) (2019) 1–14.
- [31] A. Nedic, A. Olshevsky, W. Shi, C. A. Uribe, Geometrically convergent distributed optimization with uncoordinated step-sizes, in: *Proceedings of the American Control Conference, 2017*, pp. 3950–3955.
- [32] H. Li, Q. Lü, G. Chen, T. Huang, Z. Dong, Convergence of distributed accelerated algorithm over unbalanced directed networks, *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 51 (8) (2019) 5153–5164.

- [33] J. Gao, X. W. Liu, Y. H. Dai, Y. Huang, P. Yang, Achieving geometric convergence for distributed optimization with Barzilai-Borwein step sizes, *Science China Information Sciences* 65 (4) (2022) 2021–2022.
- [34] Y. Huang, H. Liu, S. Zhou, An efficient monotone projected Barzilai-Borwein method for nonnegative matrix factorization, *Applied Mathematics Letters* 45 (2015) 12–17.
- [35] Y. Huang, H. Liu, Smoothing projected Barzilai-Borwein method for constrained non-Lipschitz optimization, *Computational Optimization and Applications* 65 (3) (2016) 671–698.
- [36] C. Tan, S. Ma, Y. H. Dai, Y. Qian, Barzilai-Borwein step size for stochastic gradient descent, in: *Advances in Neural Information Processing Systems*, 2016, pp. 685–693.
- [37] K. Ma, J. Zeng, J. Xiong, Q. Xu, X. Cao, W. Liu, Y. Yao, Stochastic non-convex ordinal embedding with stabilized Barzilai-Borwein step size, in: *32nd AAAI Conference on Artificial Intelligence, AAAI, 2018*, pp. 3738–3745.
- [38] J. Xu, S. Zhu, C. Y. Soh, L. Xie, Convergence of asynchronous distributed gradient methods over stochastic networks, *IEEE Transactions on Automatic Control* 63 (2) (2017) 434–448.
- [39] Q. Lü, H. Li, D. Xia, Geometrical convergence rate for distributed optimization with time-varying directed graphs and uncoordinated step-sizes, *Information Sciences* 422 (2018) 516–530.
- [40] K. Yuan, B. Ying, X. Zhao, A. H. Sayed, Exact diffusion for distributed optimization and learning—Part I: algorithm development, *IEEE Transactions on Signal Processing* 67 (3) (2019) 708–723.
- [41] H. Li, Q. Lü, T. Huang, Convergence analysis of a distributed optimization algorithm with a general unbalanced directed communication network,

IEEE Transactions on Network Science and Engineering 6 (3) (2018) 237–248.

- [42] R. Horn, C. R. Johnson, Matrix analysis, 2nd ed., Cambridge University Press, 2012.
- [43] H. Li, Q. Lü, T. Huang, Distributed projection subgradient algorithm over time-varying general unbalanced directed graphs, IEEE Transactions on Automatic Control 64 (3) (2018) 1309–1316.
- [44] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Convex optimization: Algorithms and complexity, Foundations and Trends in Machine Learning 8 (3-4) (2015) 231–357.
- [45] D. Dua, C. Graff, UCI (University of California Irvine) Machine Learning repository (2019).