mathematics

MDPI

Article

A Distributed Optimization Accelerated Algorithm with Uncoordinated Time-Varying Step-Sizes in an Undirected Network

Yunshan Lü [1,2], Hailing Xiong [1,3,*], Hao Zhou [1] and Xin Guan [1]

[1] Database and Artificial Intelligence Laboratory, College of Computer and Information Science, Southwest University, Chongqing 400715, China; lvyunshan@email.swu.edu.cn (Y.L.); zhouhao19@email.swu.edu.cn (H.Z.); guanxin2020@email.swu.edu.cn (X.G.)
[2] College of Big Data and Software, Chongqing College of Mobile Communication, Chongqing 401520, China
[3] Business College, Southwest University, Chongqing 402460, China
* Correspondence: xionghl@swu.edu.cn; Tel.: +86-139-9645-3236

Abstract: In recent years, significant progress has been made in the field of distributed optimization algorithms. This study focused on the distributed convex optimization problem over an undirected network. The target was to minimize the average of all local objective functions known by each agent while each agent communicates necessary information only with its neighbors. Based on the state-of-the-art algorithm, we proposed a novel distributed optimization algorithm, when the objective function of each agent satisfies smoothness and strong convexity. Faster convergence can be attained by utilizing Nesterov and Heavy-ball accelerated methods simultaneously, making the algorithm widely applicable to many large-scale distributed tasks. Meanwhile, the step-sizes and accelerated momentum coefficients are designed as uncoordinate, time-varying, and nonidentical, which can make the algorithm adapt to a wide range of application scenarios. Under some necessary assumptions and conditions, through rigorous theoretical analysis, a linear convergence rate was achieved. Finally, the numerical experiments over a real dataset demonstrate the superiority and efficacy of the novel algorithm compared to similar algorithms.

Keywords: distributed convex optimization; accelerated method; uncoordinated; undirected network; linear convergence

Citation: Lü, Y.; Xiong, H.; Zhou, H.; Guan, X. A Distributed Optimization Accelerated Algorithm with Uncoordinated Time-Varying Step-Sizes in an Undirected Network. *Mathematics* **2022**, *10*, 357. https://doi.org/10.3390/math10030357

Academic Editor: David Greiner

Received: 12 December 2021
Accepted: 21 January 2022
Published: 25 January 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

1. Introduction

In recent years, with the rapid development of artificial intelligence, big data, etc., there has been much attention to distributed optimization problems in multi-agent systems. As one of the most important fields, distributed optimization methods have gained significant growing interest due to the widespread applications in science and engineering areas such as the transmission of information in wireless sensor networks [1–3], the collaboration of vehicles in formation control [4,5], speeding up the optimization process in distributed machine learning [6,7], distributed resource allocation in smart-grid networks [8–10], distributed control in nonlinear dynamical systems [11,12], etc. Specifically, a distributed optimization framework can avoid the establishment of long-distance communication between agents while providing better load balancing for the network. Compared to traditional centralized optimization, agents in a multi-agent system communicate information only with their neighbors for distributed optimization. At the same time, the local objective function of each agent is known only by itself.

Literature Review: Since the DGD (Distributed gradient descent) algorithm was proposed by Nedic [13] for solving distributed convex problem in multiagent systems, great progress has been made in the distributed optimization field. Especially, the distributed first-order methods have attracted many researchers' attention. Based on consensus theory [14] and gradient-descent technology, diminishing step-sizes were introduced into the algorithm

DGD [13], which made the algorithm converge to the exact optimal solution but with a sublinear rate. When there were constraints of a decision variable, by utilizing the projection method, Sundhar [15] proposed a stochastic subgradient projection algorithm. Similar to [13,15], refs. [16–18] also employed the diminishing step-sizes, and these algorithms could converge linearly. However, diminishing step-sizes will lead to a much slower convergence rate. Then, the distributed algorithms with constant step-sizes were developed in [19–32] to overcome the shortcoming. The algorithm EXTRA [19] (Extra: An exact first-order algorithm for decentralized consensus optimization) and its improvement [20–22] modified the update rule of DGD by taking the difference of two consecutive iterations of formulas. Compared to DGD, the linear convergence rate can be verified in EXTRA, and even the step-size was fixed to a constant; EXTRA was more stable, but two weight matrices in EXTRA must obey strict conditions called the Mixing Matrix. A different type of distributed optimization algorithm HSADO (harnessing smoothness to accelerate distributed optimization) was proposed by Qu and Li [26] when the local objective functions were convex and smooth. HSADO adopted a gradient-tracking mechanism, which replaced the gradient term in DGD with a tracking gradient that was the gradient estimation of the average gradient of the whole network. If the step-sizes were set to constants, HSADO also can converge to the optimal solution linearly. Based on HSADO, researchers modified to adapt different scenarios, such as time-varying networks [27,28] and node-varying [29,30] and accelerated methods [31,32]. Further, researchers studied the primal-dual method in distributed optimization by utilizing the Augmented Lagrangian function; the original problem was reformulated in a dual problem. It has been demonstrated that EXTRA was equivalent to the algorithms in [33,34] by introducing dual variables, and [27] also provided a primal-dual interpretation for HSADO. Recently, the primal-dual algorithm UG (A unification and generalization of exact distributed first-order methods) proposed in [35] unified and generalized the methods EXTRA and HSADO, while it also converged linearly.

Motivations: Among these studies, EXTRA, HSADO, and UG are most related to our research. The algorithm UG can be regard as a generalization and unification of DGD, EXTRA, and HSADO. However, each local objective function of the agent in the network requires to be twice continuously differentiable, which is a rigorous condition in actual scenarios. In order to obtain linear convergence, fixed constant step-sizes were frequently adopted in distributed optimization algorithms such as [19,26,35], etc. Unfortunately, uncoordinated step-sizes for different agents are required rather than the same constant step-sizes. This situation was first studied in [36], in which an augmented distributed-gradient method was proposed, but it converged sublinearly. Then, by employing uncoordinated step-sizes, Lü [30] and Jakovetic [28] both established a global linear convergence of their algorithms in time-varying undirected and directed networks, respectively. To endow more independence, time-varying and nonidentical step-sizes of each agent were studied. A primal-dual fixed-point algorithm with nonidentical step-sizes was proposed by Li [37] when the object function of each agent was twice differentiable and nonsmooth. Xin [32] also adopted nonidentical step-sizes in a directed network. With a more relaxed step-size and network topolog, a distributed primal-dual optimization method in [38] was proposed by utilizing time-varying step-sizes, which was proved to converge linearly. Until now, to the best knowledge of the authors, no related studies for the widely used algorithm UG with uncoordinated, time-varying, and nonidentical step-sizes in an undirected network were studied. Recently, as optimization processes of large-scale tasks such as deep learning are getting slower, the convergence rate of distributed optimization algorithms need further improvement. With the help of Nesterov [39] and Heavy-ball [40] accelerated methods, the convergence rate of distributed optimization algorithms can be improved. In [32], a Heavy-ball distributed accelerated method with gradient-tracking technology was proposed to accelerate the well-known row-stochastic and column-stochastic algorithm [41]. In [31,42], a better convergence rate was shown by utilizing the Nesterov accelerated method. Moreover, both the Heavy-ball and Nesterov accelerated methods were introduced to improve the convergence rate in directed networks for machine learning in [43]. For the widely used algorithm UG, it is challenging to study whether the simultaneous inclusion of Heavy-

ball and Nesterov momentum can bring about a faster convergence rate in large-scale computing and communication tasks.

Statement of Contributions: Throughout this article, we mainly focus on the application of distributed convex optimization method over an undirected network. We propose a novel distributed optimization algorithm with uncoordinated, time-varying, and nonidentical step-sizes and accelerated momentum terms, which has a faster linear convergence rate and can apply to more scenarios. To summarize, three contributions are as follows:

- Based on the distributed optimization methods [19,26,35], we designed and discussed a faster distributed optimization accelerated algorithm, named UGNH (UG with Nesterov and Heavy-ball accelerated methods), which solves the distributed convex problems over an undirected network. In particular, the momentum with the Nesterov and Heavy-ball methods together improve the convergence rate, which can be seen in the numerical experiments.
- Compared to related algorithms, in our algorithm, not only the step-sizes but the coefficients of momentum terms (for convenience, we call them coefficients for short later) are uncoordinated, time-varying, and nonidentical, which are locally chosen for each agent. Through convergence analysis, the step-sizes and coefficients are more flexible than most existing methods. Meanwhile, if the local objective functions satisfy the conditions that are smooth and strongly convex, we can obtain an upper bound of step-sizes and coefficients. Under the upper bounds, the sequences generated by UGNH converge to the exact optimal solutions linearly.
- In contrast to related algorithms, the upper bounds of the largest step-sizes and coefficients of UGNH are more relaxed, which only depend on the parameters of objective functions and the topology of the network. Meanwhile, there can be zero step-sizes and coefficients (not all) among agents.

Organization: The rest of this article is arranged as follows. In Section 2, we describe the distributed problem and provide some necessary assumptions. In Section 3, we discuss the development of relevant distributed optimization algorithms and two classical accelerated methods and then propose a new distributed accelerated algorithm. Convergence analysis is detailed in Section 4. In Section 5, numerical experiments are provided to demonstrate the superiority and efficiency of our algorithm. Finally, Section 6 concludes this article and provides some research directions for the future.

Basic Notation: Throughout the rest of this article, unless otherwise specified, all vectors are considered as column vectors, and n is the number of agents in network. The real-number set, the natural-number set, and the m-dimensional real column vector are denoted by $\mathbb{R}$, $\mathbb{N}$, and $\mathbb{R}^m$, respectively. The subscript notations $i, j \in \{1, 2, \cdots, n\}$ represent the indices of the agents, while the superscript notation t represents an index for the iteration step, e.g., x_i^t represents the ith agent's decision variable at the jth iteration; $0_n \in \mathbb{R}^n$,$1_n \in \mathbb{R}^n$, and $I_n \in \mathbb{R}^{n\times n}$ denote an n-dimensional zero vector, one vector, and an identity matrix, respectively. For a matrix P, p_{ij} denotes the element at the i-th row and the j-th column of P, while its spectral radius and spectral norm are defined as $\rho(P)$ and $\|P\|$, respectively. Similarly, $\|x\|$ denotes the 2-norm for vector x. The transpose of a vector x and a matrix P are denoted by x^T and P^T, respectively. For a vector $r = [r_1, r_2, \cdots, r_n]^T$, diag$(r)$ represents a diagonal matrix, the diagonal elements of which equal to the vector r. The notation $\otimes$ represents the Kronecker product. Let $\nabla f(x) : \mathbb{R}^m \to \mathbb{R}^m$ denote the gradient of $f(x)$ at x.

2. Preliminaries

This section describes the formulation of the distributed optimization problem and some necessary basic assumptions related to network and function.

2.1. Problem Formulation

Consider an undirected network of n agents, which cooperatively solve the optimization problem written in the following form over a common variable $x \in \mathbb{R}^m$:

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

Here, each local objective function $f_i : \mathbb{R}^m \to \mathbb{R}$ with a convex property is possessed by agent i, which exchanges local information only with its neighbors. Our main target was to design a distributed optimization algorithm, a decision variable of which can linearly converge to the optimal solution that minimizes the average of all local objective functions. The optimal average objective value of problem (1) is defined as $f(\tilde{x}^*)$, where $\tilde{x}^* \in \mathbb{R}^m$ is the optimal decision variable. Then, the global optimal solution of (1) is denoted by $x^* \in \mathbb{R}^{nm}$, where $x^* = 1_n \otimes \tilde{x}^*$.

As a local copy of the global decision variable is saved at each agent, optimization problem (1) can be solved in a distributed way by iterating the decision variable. In this study, network is described as $\mathcal{G} = \{V, E\}$, where $V = \{1, 2, \cdots, n\}$ is the vertex set that represents the agents of the network, and $E = \{(i, j) | i, j \in V\}$ is the edges set. In an undirected network, an edge $(i, j) \in E$ implies that an edge $(j, i) \in E$ too. Meanwhile, agent i and agent j can exchange information with each other. Let $N_i = \{j | (i, j) \in E\} \cup \{i\}$ denote the set of all neighbors of agent i. Then, formulation (1) can be rewritten as follows:

$$\min_{x \in \mathbb{R}^{nm}} f(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x_i) \tag{2}$$

where $x = \left[x_1^T, x_2^T, \cdots, x_n^T\right]^T \in \mathbb{R}^{nm}$, $x_i = x_j$ for $\forall i, j \in V$. Recently, it has been proved that a new equality $\frac{1}{\alpha} L^{\frac{1}{2}} x = 0$ is equivalent to the consensus condition $x_i = x_j$ in [33], where α is the step-size, and $L = I - P$ is a Laplace matrix. Then, the primal-dual method can be introduced to solve (2) by utilizing the Augmented Lagrangian function, which is also a cornerstone of our algorithm.

Next, some necessary assumptions about the underlying graph and local objective functions are formalized, which are a common standard in related distributed optimization studies.

2.2. Assumptions

Assumption 1 ([35]). *The network $\mathcal{G} = \{V, E\}$ is connected, undirected, and simple. In particular, there are no self-loops of any agent and no multiple links between any two agents.*

Assumption 2 ([19]). *A non-negative symmetric doubly stochastic weight matrix $P = \{p_{ij}\} \in \mathbb{R}^{n \times n}$ is defined to represent network $\mathcal{G}$. The weight of the matrix P satisfies the following three conditions:*

- *Non-negative:* $p_{ij} = \begin{cases} > 0, j \in N_i \\ = 0, otherwise \end{cases}$
- *Symmetric:* $p_{ij} = p_{ji}$
- *Doubly stochastic:* $\sum_{i=1}^{n} p_{ij} = \sum_{j=1}^{n} p_{ij} = 1$

Assumption 3. *Each local objective function $f_i : \mathbb{R}^m \to \mathbb{R}, i \in V$ is smooth with Lipschitz constant ψ_i and strongly convex with parameter μ_i. Mathematically, there exits $\psi_i > 0$, $\mu_i \geq 0 (\sum \mu_i > 0)$, for any $x, y \in \mathbb{R}^m$ such that:*

$$\|\nabla f_i(x) - \nabla f_i(y)\| \leq \psi_i \|x - y\|$$
$$f_i(x) - f_i(y) \geq \nabla f_i(x)^T (x - y) + \frac{\mu_i}{2} \|x - y\|$$

Remark 1. *Assumption 1 ensures that each agent can directly or indirectly affect other agents in the network. Assumption 3 is a standard assumption in convergence analysis of distributed optimization methods. Especially, under the assumption of strongly convex for each function, there exists a unique global optimal solution to problem (1). Moreover, for the global objective function f, we define $\bar{\psi} = \left(\frac{1}{n}\right)\sum_{i=1}^{n}\psi_i$ as the global Lipschitz constant and $\bar{\mu} = \left(\frac{1}{n}\right)\sum_{i=1}^{n}\mu_i$ as the global strongly convex parameter.*

3. Algorithm Development

In this section, Section 3.1 describes the development of some related algorithms. Section 3.2 describes the Nesterov and Heavy-ball accelerated methods for the distributed optimization algorithm. Section 3.3 describes the proposed algorithm UGNH and the relationship between UGNH and the previous algorithms.

3.1. Related Algorithms

In this subsection, we focus on some classical algorithms DGD, EXTRA, HSADO, and UG, which are related to the proposed algorithm and give them a simple explanation.

In [13], Nedic and Ozdaglar proposed a standard distributed gradient descent method DGD. The method updated the decision variable at each agent through its neighbors and the local negative gradient's direction, as follows:

$$x_i^{t+1} = \sum_{j\in N_i} p_{ij}x_j^t - \alpha^t \nabla f_i\left(x_i^t\right) \tag{3}$$

where α^t was the step-size, which satisfied $\alpha^t > 0$, $\sum_{t=0}^{\infty}\alpha^t = \infty$, and $\sum\left(\alpha^t\right)^2 < \infty$; the matrix P satisfied Assumption 2. The variable x_i^t stored in the agents is the local estimate of x at the t-th iteration. It was proved that sequences generated by DGD cannot converge to the exact optimal solution x^* when employing a fixed step-size, i.e., $\alpha^t = \alpha$. By taking an appropriately diminishing step-sizes, DGD can converge accurately, but the convergence rate was sublinear.

To acquire linear convergence, Shi [19] proposed a new method EXTRA by modifying the update rule of DGD (3). There were two steps performed as follows:

$$x_i^1 = \sum_{j\in N_i} p_{ij}x_j^0 - \alpha\nabla f_i\left(x_i^0\right) \tag{4}$$

$$x_i^{t+1} = x_i^t + \sum_{j\in N_i} p_{ij}x_j^t - \sum_{j\in N_i}\tilde{p}_{ij}x_j^{t-1} - \alpha\left(\nabla f_i\left(x_i^t\right) - \nabla f_i\left(x_i^{t-1}\right)\right) \tag{5}$$

where the step-size $\alpha > 0$ was a constant, the matrix P satisfied Assumption 2, while $\tilde{P} = \frac{I+P}{2}$ was appropriate. Compared to DGD (3), an initial condition (4) and one more iteration (5) were added. Notably, the step-size was a constant, but EXTRA can linearly converge to the exact optimal solution as long as the step-size was chosen appropriately.

Based on DGD (3), Qu and Li [26] proposed a novel distributed algorithm HSADO by using gradient-tracking technology. An auxiliary variable z_i^t was introduced to estimate the network-wide gradient average $\frac{1}{n}\sum_{i=1}^{n}\nabla f_i\left(x_i^t\right)$ at the t-th iteration for agent i. As a result, the gradient contribution $-\alpha\nabla f_i\left(x_i^t\right)$ in (3) was replaced by $-\alpha z_i^t$. The specific updating rules were as follows:

$$x_i^{t+1} = \sum_{j\in N_i} p_{ij}x_j^t - \alpha z_i^t \tag{6}$$

$$z_i^{t+1} = \sum_{j\in N_i} p_{ij}z_j^t + \nabla f_i\left(x_i^{t+1}\right) - \nabla f_i\left(x_i^t\right) \tag{7}$$

where the step-size $\alpha > 0$ was a constant, and the matrix P satisfied Assumption 2. Under the previous assumptions, initialized with $x_i^0 \in \mathbb{R}^m$ and $z_i^0 = \nabla f_i\left(x_i^0\right)$, a global linear convergence rate could be gained when choosing an appropriate fixed step-size.

Recently, a novel distributed optimization algorithm UG was proposed in [35], which used the primal-dual method to solve the equivalent problem (2). Through tuning parameters, the algorithm subsumed the well-known algorithm EXTRA and HSADO. Updating rules were as follows:

$$x_i^{t+1} = \sum_{j \in N_i} p_{ij} x_j^t - \alpha \left(\nabla f_i(x_i^t) + z_i^t \right) \tag{8}$$

$$z_i^{t+1} = z_i^t - \sum_{j \in N_i} l_{ij} \left(\nabla f_j\left(x_j^t\right) + z_j^t - \sum_{q \in N_j} k_{jq} x_q^t \right) \tag{9}$$

where the step-size $\alpha > 0$ was a constant; x_i^t was primal variable; and z_i^t was dual variable, which were initialized to $x_i^0 \in \mathbb{R}^m$ and $z_i^0 = 0_m$, respectively. For using more-compact notation, we defined $P = \left\{p_{ij}\right\}$, $\mathbb{L} = \left\{l_{ij}\right\}$, and $\mathcal{K} = \left\{k_{ij}\right\}$. The matrix $\mathbb{L} = I_n - P$ and the matrix $\mathcal{K} \in \mathbb{R}^{n \times n}$ are symmetric with the property that there exists some constant λ such that $\mathcal{K} 1_n = \lambda 1_n$.

By analyzing when the matrix $\mathcal{K}$ is chosen properly, the algorithm UG is equivalent to: (1) EXTRA, when $\mathcal{K} = \frac{1}{\alpha} P$ and (2)HSADO, when $\mathcal{K} = 0_n$. For others, $\mathcal{K} = \frac{\bar{\mu}+\bar{\psi}}{2} I_n$ and $\mathcal{K} = \frac{\bar{\mu}+\bar{\psi}}{1+\lambda_n} P$ (λ_n is the smallest eigenvalue of matrix P) are appropriate for $\mathcal{K} = kI_n$ and $\mathcal{K} = kP$, respectively. There are no extra computational variables and no communication relationships with other agents in the formula $\mathcal{K}$, so (8) and (9) are easy to implement. As UG unifies and generalizes previous methods, we mainly focused on it.

3.2. Distributed Accelerated Methods

In this section, centralized Nesterov and Heavy-ball accelerated methods will be introduced. With them, many distributed optimization algorithms can converge faster.

For the gradient-descent algorithm, i.e., $x_i^{t+1} = x_i^t - \alpha \nabla f_i(x_i^t)$, the best achievable convergence is $O\left(\left(\frac{\kappa-1}{\kappa+1}\right)^t\right)$; $\kappa = \frac{\bar{\psi}}{\bar{\mu}}$ denotes the condition number of the objective function. If $\bar{\psi}$ is much larger than $\bar{\mu}$ so that κ is large, then the gradient descent becomes quite slow. To accelerate the gradient descent, Polyak [40] proposed a method called Heavy-ball for updating decision variable. The specifics was as follows:

$$x_i^{t+1} = x_i^t - \alpha \nabla f_i(x_i^t) + \gamma \left(x_i^t - x_i^{t-1} \right) \tag{10}$$

where γ was the momentum-accelerated coefficient, and the term $\gamma \left(x_i^t - x_i^{t-1} \right)$ was used to accelerate the convergence of the decision variable. It had been proved that under the appropriate step-size α and the coefficient γ, the momentum-accelerated method could achieve a convergence rate of $O\left(\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^t\right)$, which was obviously faster.

Inspired by conjugate gradient methods [44], history gradient information can improve the convergence rate for distributed first-order optimization algorithms. Nesterov proposed a method called CNGD [40] (Centralised Nesterov Gradient Descent method) as follows:

$$x_i^{t+1} = y_i^t - \alpha \nabla f_i(y_i^t) \tag{11}$$

$$y_i^{t+1} = x_i^{t+1} + \gamma \left(x_i^{t+1} - x_i^t \right) \tag{12}$$

where $\alpha = \sqrt{\frac{\bar{\mu}}{\bar{\psi}}}$, $\gamma = \frac{\sqrt{\bar{\psi}}-\sqrt{\bar{\mu}}}{\sqrt{\bar{\psi}}+\sqrt{\bar{\mu}}}$. It had been proved that CNGD achieved the best convergence rate among all centralized gradient methods within first-order algorithms. Under the previous assumptions, CNGD achieved a faster convergence rate $O\left(\left(1 - \sqrt{\frac{\bar{\mu}}{\bar{\psi}}}\right)^t\right)$, compared to the CGD's convergence rate $O\left(\left(1 - \frac{\bar{\mu}}{\bar{\psi}}\right)^t\right)$.

It is notable that the two accelerated methods have been adapted in many distributed algorithms, such as [42,43], etc. In this study, we devoted ourselves to studying the two accelerated methods on UG.

3.3. The Proposed Algorithm

The recent studies [35,42] are the most relevant to our work. Based on these works, considering that the Nesterov and Heavy-ball accelerated methods are very helpful for achieving a faster convergence, we added them into UG simultaneously. Meanwhile, in order to apply in many more scenarios, the step-sizes and coefficients were designed as uncoordinated, time-varying, and nonidentical. Combining together, we propose a new distributed optimization algorithm named UGNH as follows:

$$x_i^{t+1} = \sum_{j=1}^{n} p_{ij} y_j^t - \alpha_i^t \left(\nabla f_i\left(y_i^t\right) + z_i^t\right) + \gamma_i^t \left(x_i^t - x_i^{t-1}\right) \tag{13}$$

$$y_i^{t+1} = x_i^{t+1} + \gamma_i^t \left(x_i^{t+1} - x_i^t\right) \tag{14}$$

$$z_i^{t+1} = z_i^t - \sum_{j=1}^{n} l_{ij} \left(\nabla f_j\left(y_j^t\right) + z_i^t - \sum_{q=1}^{n} k_{jq} y_q^t\right) \tag{15}$$

where $i, j \in V$, $t \in \mathbb{N}$, the step-sizes $\alpha_i^t > 0$, and accelerated momentum coefficients $\gamma_i^t \geq 0$ are uncoordinated, time-varying, and nonidentical, which are locally chosen at each agent. At the t-th iteration, each agent stores three variables: the primal decision variable $x_i^t \in \mathbb{R}^m$, the temporary variable $y_i^t \in \mathbb{R}^m$, and the dual variable $z_i^t \in \mathbb{R}^m$, which start with initial states: $x_i^0 \in \mathbb{R}^m$, $y_i^0 \in \mathbb{R}^m$ and $z_i^0 = 0_m$. The update of UGNH at each agent i is formally described in Algorithm 1.

Algorithm 1 The update of the algorithm UGNH at each agent i

Initialization: each agent starts with: $x_i^0 \in \mathbb{R}^m$, $y_i^0 \in \mathbb{R}^m$ and $z_i^0 = 0_m$.
for $t = 0, 1, 2, \cdots$ **do**
  Update the primal decision variable x_i as follows:
  $x_i^{t+1} = \sum_{j=1}^{n} p_{ij} y_j^t - \alpha_i^t \left(\nabla f_i\left(y_i^t\right) + z_i^t\right) + \gamma_i^t \left(x_i^t - x_i^{t-1}\right)$
  Update the temporary variable y_i^{t+1} as follows:
  $y_i^{t+1} = x_i^{t+1} + \gamma_i^t \left(x_i^{t+1} - x_i^t\right)$
  for $j = 1, 2, \cdots, n$ **do**
    for $q = 1, 2, \cdots, n$ **do**
      Calculate $z_{temp} = \sum_{j=1}^{n} l_{ij} \left(\nabla f_j\left(y_j^t\right) + z_i^t - \sum_{q=1}^{n} k_{jq} y_q^t\right)$
    end for
    Update the dual variable z_i as follows:
    $z_i^{t+1} = z_i^t - z_{temp}$
  end for
end for

It is clear that UGNH is a primal-dual method; $\gamma_i^t \left(x_i^t - x_i^{t-1}\right)$ is the Heavy-ball accelerated term in (13), (14) is the Nesterov accelerated term, and (15) is the dual variable iteration. It also can be easy to verify that UGNH is equivalent to UG if $\alpha_i^t = \alpha$, $\gamma_i^t = 0_m$. Further, it can be equal to EXTRA and HSADO if the matrix $\mathcal{K}$ is chosen properly.

Remark 2. *For the sake of compaction and brevity, let the dimension $m = 1$. Other multiple dimensions can be similarly proved.*

As a result, we define: $x^t = \left[x_1^t, x_2^t, \cdots, x_n^t\right]^T \in \mathbb{R}^n$, $y^t = \left[y_1^t, y_2^t, \cdots, y_n^t\right]^T \in \mathbb{R}^n$, $z^t = \left[z_1^t, z_2^t, \cdots, z_n^t\right]^T \in \mathbb{R}^n$ *and* $\nabla F\left(y^t\right) = \left[\nabla f_1\left(y_1^t\right), \nabla f_2\left(y_2^t\right), \cdots, \nabla f_n\left(y_n^t\right)\right]^T \in \mathbb{R}^n$, *other*

notations latter used are defined as before. Then, UGNH can be compactly reformulated in a martix form as follows:

$$x^{t+1} = Py^t - \Gamma_\alpha^t\left(\nabla F(y^t) + z^t\right) + \Gamma_\gamma^t\left(x^t - x^{t-1}\right) \tag{16}$$

$$y^{t+1} = x^{t+1} + \Gamma_\gamma^t\left(x^{t+1} - x^t\right) \tag{17}$$

$$z^{t+1} = z^t - \mathbb{L}\left(\nabla F(y^t) + z^t - \mathcal{K}y^t\right) \tag{18}$$

where $\alpha^t = \left[\alpha_1^t, \alpha_2^t, \cdots, \alpha_n^t\right]^T \in \mathbb{R}^n$ *and* $\gamma^t = \left[\gamma_1^t, \gamma_2^t, \cdots, \gamma_n^t\right]^T \in \mathbb{R}^n$ *represent step-sizes and coefficients, respectively. Furthermore,we define* $\Gamma_\alpha^t = diag\left(\alpha^t\right) \in \mathbb{R}^{n\times n}$ *and* $\Gamma_\gamma^t = diag\left(\gamma^t\right) \in \mathbb{R}^{n\times n}$.

4. Convergence Analysis

This section analyzes in detail the linear convergence of decision variable sequences generated by UGNH when step-sizes and coefficients are chosen properly. First, we define some notations that may frequently be used later.

$$\bar{x}^t = \frac{1}{n}1_n^T x^t, \bar{z}^t = \frac{1}{n}1_n^T z^t, J_n = \frac{1}{n}1_n 1_n^T, \tilde{\psi} = \max_{i\in V}\{\psi_i\},$$
$$a = \|P - J_n\|, b = \|I_n - J_n\|, c = \|P - I_n\|, d = \|\mathbb{L}\|(\tilde{\psi} + \|\mathcal{K}\|)$$

Moreover, considering that the step-sizes and coefficients are uncoordinated, time-varying, and nonidentical, there are many possible numerical values that may be difficult to handle. By employing a small trick, we only studied the supremum and infimum of the step-sizes and coefficients. The specific definitions are as follows:

$$\alpha_{\max} = \sup_{t\geq 0}\max_{i\in V}\{\alpha_i^t\}, \alpha_{\min} = \inf_{t\geq 0}\min_{i\in V}\{\alpha_i^t\}, \tilde{\gamma} = \sup_{t\geq 0}\max_{i\in V}\{\gamma_i^t\}$$

In addition, let $\xi_\alpha = \alpha_{\max} - \alpha_{\min}$ be the difference between $\alpha_{\max}$ and $\alpha_{\min}$, and let $\Phi = \frac{\alpha_{\max}}{\alpha_{\min}}$ be the condition number.

Before giving the main results, we introduce some helpful supporting lemmas for the convergence analysis.

4.1. Supporting Lemmas

Lemma 1 ([26]). *Under Assumption 3, the global objective function* f *is* $\bar{\psi}$*-smooth and* $\bar{\mu}$*-strongly convex. For any* $x \in \mathbb{R}$ *and* $0 < \alpha < \frac{2}{\bar{\psi}}$*, we have:*

$$\|x - \alpha\nabla f(x) - x^*\| \leq \zeta\|x - x^*\|$$

where $\zeta = \max\{|1 - \bar{\psi}\alpha|, |1 - \bar{\mu}\alpha|\}$.

Lemma 2 ([19]). *Assumption null* $\{I_n - P\} = span\{1\}$*, matrix* P *satisfies Assumption 2,* x^* *is the optimal solution when* x^* *satisfies the following conditions:*

- $x^* = Px^*$ *(consensus)*
- $1_n^T\nabla F(x^*) = 0$ *(optimality)*

Lemma 3 ([32]). *Assume that a matrix* $P \in \mathbb{R}^{n\times n}$ *and a vector* $\varepsilon \in \mathbb{R}^n$ *are non-negative and positive, respectively; if* $P\varepsilon < \varrho\varepsilon$ *with* $\varrho > 0$*, we have* $\rho(P) < \varrho$.

4.2. Main Results

In this section, the linear-convergence analysis of the proposed algorithm is carried out in detail. Similar to relevant studies, we mainly focus on the following four mathematical expressions at the $(t+1)$-th iteration: $x^{t+1} - 1_n \otimes \bar{x}^{t+1}$, $1_n \otimes \bar{x}^{t+1} - x^*$, $x^{t+1} - x^t$, and $z^{t+1} - z^*$. For convenience, let Ξ_1^{t+1}, Ξ_2^{t+1}, Ξ_3^{t+1}, and Ξ_4^{t+1} represent the four expressions, respectively. Among them, by introducing the norm, $\|\Xi_1^{t+1}\|$ is described as a consensus

violation, $\|\Xi_2^{t+1}\|$ as an optimal residual, $\|\Xi_3^{t+1}\|$ as a state difference, and $\|\Xi_4^{t+1}\|$ as a dual error.

Next, we spared no effort to bound the four norm expressions at the $(t+1)$-th iteration through their estimates at the t-th iteration in terms of linear combinations. Subsequently, based on Assumptions 1–3, we established a linear inequalities system for convergence analysis. In what follows, consensus violation $\|\Xi_1^{t+1}\|$ is bounded first.

Lemma 4. *$\forall\, t > 0$, the following inequality holds:*

$$\|\Xi_1^{t+1}\| \leq (a + b\alpha_{\max}\tilde{\psi})\|\Xi_1^t\| + b\alpha_{\max}\tilde{\psi}\|\Xi_2^t\| + (b\alpha_{\max}\tilde{\psi} + a + b)\tilde{\gamma}\|\Xi_3^t\| + b\alpha_{\max}\|\Xi_4^t\| \quad (19)$$

Proof of Lemma 4. Considering (16) and (17), we have:

$$x^{t+1} = Px^t - \Gamma_\alpha^t\big(\nabla F(y^t) + z^t\big) + P\Gamma_\gamma^{t-1}\Xi_3^t + \Gamma_\gamma^t\Xi_3^t \quad (20)$$

Note that $(I_n - J_n)x^{t+1} = \Xi_1^{t+1}$, $(I_n - J_n)P = P - J_n$ and $(P - J_n)1_n = 0_n$, multiplying $(I_n - J_n)$ on both sides of (20), then:

$$\begin{aligned}\Xi_1^{t+1} =& (P - J_n)\Xi_1^t + (I_n - J_n)\Gamma_\gamma^t\Xi_3^t - (I_n - J_n)\Gamma_\alpha^t\big(\nabla F(y^t) - \nabla F(x^*)\big)\\ &- (I_n - J_n)\Gamma_\alpha^t\big(z^t + \nabla F(x^*)\big) + (P - J_n)\Gamma_\gamma^{t-1}\Xi_3^t\end{aligned} \quad (21)$$

Based on the fact $z^t - z^* = z^t + \nabla F(x^*)$ [35] and Assumption 3, taking the norm on both sides of (21), then:

$$\begin{aligned}\|\Xi_1^{t+1}\| \leq& \|P - J_n\|\|\Xi_1^t\| + \|I_n - J_n\|\alpha_{\max}\tilde{\psi}\big(\|\Xi_1^t\| + \|\Xi_2^t\|\big) + \|I_n - J_n\|\alpha_{\max}\tilde{\psi}\tilde{\gamma}\|\Xi_3^t\|\\ &+ \|I_n - J_n\|\alpha_{\max}\|\Xi_4^t\| + \|P - J_n\|\tilde{\gamma}\|\Xi_3^t\| + \|I_n - J_n\|\tilde{\gamma}\|\Xi_3^t\|\end{aligned} \quad (22)$$

Recalling the definition of a and b, then:

$$\begin{aligned}\|\Xi_1^{t+1}\| \leq& a\|\Xi_1^t\| + b\alpha_{\max}\tilde{\psi}\|\Xi_1^t\| + b\alpha_{\max}\tilde{\psi}\|\Xi_2^t\| + b\alpha_{\max}\tilde{\psi}\tilde{\gamma}\|\Xi_3^t\|\\ &+ b\alpha_{\max}\|\Xi_4^t\| + (a + b)\tilde{\gamma}\|\Xi_3^t\|\end{aligned} \quad (23)$$

Rearranging the terms in (23), the result in Lemma 4 is obtained. □

Lemma 5. *$\forall\, t > 0$, the following inequality holds:*

$$\|\Xi_2^{t+1}\| \leq (\alpha_{\max}\tilde{\psi} + \xi_\alpha\tilde{\psi})\|\Xi_1^t\| + (\zeta + \xi_\alpha\tilde{\psi})\|\Xi_2^t\| + (\alpha_{\max}\tilde{\psi} + \xi_\alpha\tilde{\psi} + 2)\tilde{\gamma}\|\Xi_3^t\| + \xi_\alpha\|\Xi_4^t\| \quad (24)$$

Proof of Lemma 5. Multiplying J_n on both of (16), and substituting $y^t = x^t + \Gamma_\gamma^{t-1}\Xi_3^t$, we have:

$$J_n x^{t+1} = J_n x^t - J_n\Gamma_\alpha^t\big(\nabla F(y^t) + z^t\big) + J_n\Gamma_\gamma^{t-1}\Xi_3^t + J_n\Gamma_\gamma^t\Xi_3^t \quad (25)$$

To get the related terms, recalling the fact that $\bar{z}^{t+1} = \bar{z}^t = \cdots = \bar{z}^0 = 0$ (e.g., $J_n z^t = 0$) in [35], we add some useful items and delete them in (25) as follows:

$$\begin{aligned}J_n x^{t+1} =& J_n x^t - \alpha_{\max}J_n\nabla F(J_n x^t) + \alpha_{\max}J_n\big(\nabla F(J_n x^t) - \nabla F(y^t)\big) + J_n\Gamma_\gamma^{t-1}\Xi_3^t + J_n\Gamma_\gamma^t\Xi_3^t\\ &+ J_n\big(1_n \otimes \alpha_{\max} - \Gamma_\alpha^t\big)\big(\nabla F(y^t) - \nabla F(x^*)\big) + J_n\big(1_n \otimes \alpha_{\max} - \Gamma_\alpha^t\big)\big(z^t + \nabla F(x^*)\big)\end{aligned} \quad (26)$$

By applying $\nabla f(x) = \frac{1}{n}1_n\nabla F(x)$, subtracting x^* on the sides of (26), we then obtain:

$$\begin{aligned}\Xi_2^{t+1} =& 1_n\big(\bar{x}^t - \tilde{x}^* - \alpha_{\max}\nabla f(\bar{x}^t)\big) + \alpha_{\max}J_n\big(\nabla F(J_n x^t) - \nabla F(y^t)\big) + J_n\Gamma_\gamma^{t-1}\Xi_3^t + J_n\Gamma_\gamma^t\Xi_3^t\\ &+ J_n\big(1_n \otimes \alpha_{\max} - \Gamma_\alpha^t\big)\big(\nabla F(y^t) - \nabla F(x^*)\big) + J_n\big(1_n \otimes \alpha_{\max} - \Gamma_\alpha^t\big)\big(z^t + \nabla F(x^*)\big)\end{aligned} \quad (27)$$

Taking the norm on both sides of (27) and using Lemma 1, then:

$$\begin{aligned}\|\Xi_2^{t+1}\| \leq & \zeta\|\Xi_2^t\| + \alpha_{\max}\|J_n\|\tilde{\psi}\|\Xi_1^t - \Gamma_\gamma^{t-1}\Xi_3^t\| + \|J_n\|(\alpha_{\max}-\alpha_{\min})\tilde{\psi}\|y^t - x^*\| \\ & + \|J_n\|(\alpha_{\max}-\alpha_{\min})\|\Xi_4^t\| + 2\|J_n\|\tilde{\gamma}\|\Xi_3^t\| \\ \leq & \zeta\|\Xi_2^t\| + \alpha_{\max}\tilde{\psi}\|\Xi_1^t\| + \alpha_{\max}\tilde{\psi}\tilde{\gamma}\|\Xi_3^t\| + \xi_\alpha\tilde{\psi}\|\Xi_1^t\| + \xi_\alpha\tilde{\psi}\|\Xi_2^t\| \\ & + \xi_\alpha\tilde{\psi}\tilde{\gamma}\|\Xi_3^t\| + \xi_\alpha\|\Xi_4^t\| + 2\tilde{\gamma}\|\Xi_3^t\| \end{aligned} \tag{28}$$

Rearranging the terms in (28), the desired results can be obtained. □

Lemma 6. *$\forall\, t > 0$, the following inequality holds:*

$$\|\Xi_3^{t+1}\| \leq (c + \alpha_{\max}\tilde{\psi})\|\Xi_1^t\| + \alpha_{\max}\tilde{\psi}\|\Xi_2^t\| + (\alpha_{\max}\tilde{\psi}\tilde{\gamma} + 2\tilde{\gamma})\|\Xi_3^t\| + \alpha_{\max}\|\Xi_4^t\| \tag{29}$$

Proof of Lemma 6. Substituting $y^t = x^t + \Gamma_\gamma^{t-1}\Xi_3^t$ in (16), then subtracting x^t on both sides, then:

$$\begin{aligned}\Xi_3^{t+1} = & P\left(x^t + \Gamma_\gamma^{t-1}\Xi_3^t\right) - x^t - \Gamma_\alpha^t\left(\nabla F\left(y^t\right) + z^t\right) + \Gamma_\gamma^t\Xi_3^t \\ = & (P - I_n)\Xi_1^t - \Gamma_\alpha^t\left(\nabla F\left(y^t\right) - \nabla F(x^*)\right) + \Gamma_\alpha^t\Xi_4^t + \left(P\Gamma_\gamma^{t-1} + \Gamma_\gamma^t\right)\Xi_3^t \end{aligned} \tag{30}$$

The second equality is based on $(P - I_n)1_n = 0_n$; recalling the definition of c and taking the norm on both sides of (30), we have:

$$\begin{aligned}\|\Xi_3^{t+1}\| \leq & c\|\Xi_1^t\| + \alpha_{\max}\tilde{\psi}\|y^t - x^*\| + \alpha_{\max}\|\Xi_4^t\| + 2\tilde{\gamma}\|\Xi_3\| \\ = & c\|\Xi_1^t\| + \alpha_{\max}\tilde{\psi}\|\Xi_1^t\| + \alpha_{\max}\tilde{\psi}\|\Xi_2^t\| + \alpha_{\max}\tilde{\psi}\tilde{\gamma}\|\Xi_3^t\| + \alpha_{\max}\|\Xi_4^t\| + 2\tilde{\gamma}\|\Xi_3^t\| \end{aligned} \tag{31}$$

Rearranging the terms in (31) , the result in Lemma 6 is obtained. □

Lemma 7. *Let Assumptions 2–3 and Lemma 2 hold. $\forall\, t > 0$, the following inequality holds:*

$$\|\Xi_4^{t+1}\| \leq d\|\Xi_1^t\| + d\|\Xi_2^t\| + d\tilde{\gamma}\|\Xi_3^t\| + a\|\Xi_4^t\| \tag{32}$$

Proof of Lemma 7. Noting $(P - J_n)1_n = 0_n$ and adding $\nabla F(x^*)$ on both sides of (18), we have:

$$\begin{aligned} z^{t+1} + \nabla F(x^*) = & z^t + \nabla F(x^*) - \mathbb{L}\left(\nabla F\left(y^t\right) + z^t - \mathcal{K}y^t\right) \\ = & P\left(z^t + \nabla F(x^*)\right) + \mathbb{L}\mathcal{K}y^t - (I_n - P)\left(\nabla F\left(y^t\right) - \nabla F(x^*)\right) \\ = & (P - J_n)\left(z^t - z^*\right) + \mathbb{L}\mathcal{K}\left(y^t - x^*\right) - \mathbb{L}\left(\nabla F\left(y^t\right) - \nabla F(x^*)\right) \end{aligned} \tag{33}$$

The third equality of (33) is from the following fact in [35] and Lemma 2:
$\bar{z}^{t+1} = \bar{z}^t = \cdots = \bar{z}^0 = 0$, $J_n\nabla F(x^*) = 0_n$ and $\mathbb{L}\mathcal{K}1_n = 0_n$.
Recalling the definition of d, taking the norm on both sides of (33),we have:

$$\|\Xi_4^{t+1}\| \leq \|P - J_n\|\|\Xi_4^t\| + (\|\mathbb{L}\|\tilde{\psi} + \|\mathbb{L}\|\|\mathcal{K}\|)\|y^t - x^*\| = a\|\Xi_4^t\| + d\|y^t - x^*\| \tag{34}$$

Substituting $y^t = x^t + \Gamma_\gamma^{t-1}\Xi_3^t$ in (34) and rearranging the terms can yield the desired result. □

With the Lemmas 4–7 above, we established the main convergence result as follows.

Theorem 1. *Suppose that Assumptions 1–3 hold. Considering the sequences $\{x^t\}$, $\{y^t\}$, and $\{z^t\}$ generating by the proposed algorithm UGNH and combining Lemmas 4–7 in a linear-inequalities system, we have:*

$$\begin{bmatrix} \|\Xi_1^{t+1}\| \\ \|\Xi_2^{t+1}\| \\ \|\Xi_3^{t+1}\| \\ \|\Xi_4^{t+1}\| \end{bmatrix} \leq \mathbb{H} \begin{bmatrix} \|\Xi_1^{t}\| \\ \|\Xi_2^{t}\| \\ \|\Xi_3^{t}\| \\ \|\Xi_4^{t}\| \end{bmatrix} \tag{35}$$

where the matrix $\mathbb{H} \in \mathbb{R}^{4\times 4}$ *is given as below:*

$$\mathbb{H} = \begin{bmatrix} a + b\alpha_{\max}\tilde{\psi} & b\alpha_{\max}\tilde{\psi} & b\alpha_{\max}\tilde{\psi}\tilde{\gamma} + a\tilde{\gamma} + b\tilde{\gamma} & b\alpha_{\max} \\ \alpha_{\max}\tilde{\psi} + \xi_\alpha\tilde{\psi} & \zeta + \xi_\alpha\tilde{\psi} & \alpha_{\max}\tilde{\psi}\tilde{\gamma} + \xi_\alpha\tilde{\psi}\tilde{\gamma} + 2\tilde{\gamma} & \xi_\alpha \\ c + \alpha_{\max}\tilde{\psi} & \alpha_{\max}\tilde{\psi} & \alpha_{\max}\tilde{\psi}\tilde{\gamma} + 2\tilde{\gamma} & \alpha_{\max} \\ d & d & d\tilde{\gamma} & a \end{bmatrix}$$

The largest step-size satisfies:

$$\alpha_{\max} < \min\left\{ \frac{\varepsilon_1 - a\varepsilon_1}{b\tilde{\psi}\varepsilon_1 + b\tilde{\psi}\varepsilon_2 + b\varepsilon_4}, \frac{\varepsilon_3 - c\varepsilon_1}{\tilde{\psi}\varepsilon_1 + \tilde{\psi}\varepsilon_2 + \varepsilon_4}, \frac{1}{\tilde{\psi}} \right\} \tag{36}$$

The maximum momentum coefficient satisfies:

$$\begin{aligned} \tilde{\gamma} < \min\Bigg\{ & \frac{\varepsilon_1 - a\varepsilon_1 - b\alpha_{\max}\tilde{\psi}\varepsilon_1 - b\alpha_{\max}\tilde{\psi}\varepsilon_2 - b\alpha_{\max}\varepsilon_4}{b\alpha_{\max}\tilde{\psi}\varepsilon_3 + a\varepsilon_3 + b\varepsilon_3}, \\ & \frac{\bar{\mu}\alpha_{\max}\varepsilon_2 - \xi_\alpha\tilde{\psi}\varepsilon_2 - \alpha_{\max}\tilde{\psi}\varepsilon_1 - \xi_\alpha\tilde{\psi}\varepsilon_1 - \xi_\alpha\varepsilon_4}{\alpha_{\max}\tilde{\psi}\varepsilon_3 + \xi_\alpha\tilde{\psi}\varepsilon_3 + 2\varepsilon_3}, \\ & \frac{\varepsilon_3 - \alpha_{\max}\tilde{\psi}\varepsilon_2 - c\varepsilon_1 - \alpha_{\max}\tilde{\psi}\varepsilon_1 - \alpha_{\max}\varepsilon_4}{\alpha_{\max}\tilde{\psi}\varepsilon_3 + 2\varepsilon_3}, \frac{\varepsilon_4 - d\varepsilon_1 - d\varepsilon_2 - a\varepsilon_4}{d\varepsilon_3} \Bigg\} \end{aligned} \tag{37}$$

And the conditional number satisfies:

$$1 \leq \Phi < \frac{\varepsilon_4 + \tilde{\psi}\varepsilon_2 + \tilde{\psi}\varepsilon_1}{\varepsilon_4 + \tilde{\psi}\varepsilon_2 + 2\tilde{\psi}\varepsilon_1 - \bar{\mu}\varepsilon_2} \tag{38}$$

where ε_1, ε_2, ε_3, and ε_4 are arbitrary constants, which obey the following picking rules:

$$\varepsilon_2 > 0, \varepsilon_1 < \frac{\bar{\mu}\varepsilon_2}{\tilde{\psi}}, \varepsilon_3 > c\varepsilon_1, \varepsilon_4 > \frac{d\varepsilon_1 + d\varepsilon_2}{1 - a} \tag{39}$$

Then, the spectral radius of the matrix $\mathbb{H}$ is strictly less than 1, i.e., $\rho(\mathbb{H}) < 1$, which is the desired result.

Proof of Theorem 1. According to Lemmas 4–7, we can immediately get the inequalities (35). Then, we provide some necessary conditions for parameters $\tilde{\psi}$, $\tilde{\gamma}$ and Φ, such that $\rho(\mathbb{H}) < 1$. Based on Lemma 3, let $\varepsilon = [\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4]^T \in \mathbb{R}^4$ be a positive vector, if $\mathbb{H}\varepsilon < \varepsilon$, then $\rho(\mathbb{H}) < 1$. According to the definition of $\mathbb{H}$ above, the inequality $\mathbb{H}\varepsilon < \varepsilon$ is equivalent to the following four inequalities:

$$(b\alpha_{\max}\tilde{\psi} + a + b)\tilde{\gamma}\varepsilon_3 < \varepsilon_1 - a\varepsilon_1 - b\alpha_{\max}\tilde{\psi}\varepsilon_1 - b\alpha_{\max}\tilde{\psi}\varepsilon_2 - b\alpha_{\max}\varepsilon_4 \tag{40}$$

$$(\alpha_{\max}\tilde{\psi} + \xi_\alpha\tilde{\psi} + 2)\tilde{\gamma}\varepsilon_3 < \varepsilon_2 - \alpha_{\max}\tilde{\psi}\varepsilon_1 - \xi_\alpha\tilde{\psi}\varepsilon_1 - \zeta\varepsilon_2 - \xi_\alpha\tilde{\psi}\varepsilon_2 - \xi_\alpha\varepsilon_4 \tag{41}$$

$$(\alpha_{\max}\tilde{\psi} + 2)\tilde{\gamma}\varepsilon_3 < \varepsilon_3 - c\varepsilon_1 - \alpha_{\max}\tilde{\psi}\varepsilon_1 - \alpha_{\max}\tilde{\psi}\varepsilon_2 - \alpha_{\max}\varepsilon_4 \tag{42}$$

$$d\tilde{\gamma}\varepsilon_3 < \varepsilon_4 - d\varepsilon_1 - d\varepsilon_2 - a\varepsilon_4 \tag{43}$$

According to Lemma 1, if $0 < \alpha_{\max} < \frac{1}{\tilde{\psi}}$, $\zeta = 1 - \bar{\mu}\alpha_{\max}$, then (41) is equivalent to the following inequality:

$$(\alpha_{\max}\tilde{\psi} + \xi_\alpha\tilde{\psi} + 2)\tilde{\gamma}\varepsilon_3 < \bar{\mu}\alpha_{\max}\varepsilon_2 - \alpha_{\max}\tilde{\psi}\varepsilon_1 - \xi_\alpha\tilde{\psi}\varepsilon_1 - \xi_\alpha\tilde{\psi}\varepsilon_2 - \xi_\alpha\varepsilon_4 \tag{44}$$

To make sure that the parameter $\tilde{\gamma}$ is positive, it implies that the right sides of (40) and (42)–(44) are positive. Immediately, we can get the following conditions:

$$\alpha_{\max} < \frac{\varepsilon_1 - a\varepsilon_1}{b\tilde{\psi}\varepsilon_1 + b\tilde{\psi}\varepsilon_2 + b\varepsilon_4} \tag{45}$$

$$\xi_\alpha < \frac{\bar{\mu}\alpha_{\max}\varepsilon_2 - \alpha_{\max}\tilde{\psi}\varepsilon_1}{\varepsilon_4 + \tilde{\psi}\varepsilon_2 + \tilde{\psi}\varepsilon_1}; \varepsilon_1 < \frac{\bar{\mu}\varepsilon_2}{\tilde{\psi}} \tag{46}$$

$$\alpha_{\max} < \frac{\varepsilon_3 - c\varepsilon_1}{\tilde{\psi}\varepsilon_1 + \tilde{\psi}\varepsilon_2 + \varepsilon_4}; \varepsilon_3 > c\varepsilon_1 \tag{47}$$

$$\varepsilon_4 > \frac{d\varepsilon_1 + d\varepsilon_2}{1 - a} \tag{48}$$

Recalling that $\xi_\alpha = \alpha_{\max} - \alpha_{\min}$, $\Phi = \frac{\alpha_{\max}}{\alpha_{\min}}$ as the conditional number, (46) further implies that :

$$1 \leq \Phi < \frac{\varepsilon_4 + \tilde{\psi}\varepsilon_2 + \tilde{\psi}\varepsilon_1}{\varepsilon_4 + \tilde{\psi}\varepsilon_2 + 2\tilde{\psi}\varepsilon_1 - \bar{\mu}\varepsilon_2} \tag{49}$$

Now, we attempt to select the proper vector $\varepsilon = [\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4]^T$ such that the parameters $\alpha_{\max}$, $\tilde{\gamma}$ and Φ are available. Based on (46)–(48), an arbitrary positive constant ε_2 is chosen first, and then we choose ε_1 from (46), finally choosing ε_3 and ε_4 from (47) and (48), respectively. Hence, according to (45) and (47), and the requirement of $0 < \alpha_{\max} < \frac{1}{\tilde{\psi}}$ in (44), the upper bound of the largest step-size $\alpha_{\max}$ shown in (36) can be obtained. Furthermore, according to (46), the upper bound of the conditional number Φ demonstrated in (30) can be obtained. Besides, the upper bound of the maximum coefficient $\tilde{\gamma}$ can yield from (40) and (42)–(44). Above all, the proof is finished. □

Remark 3. *According to Theorem 1, a linear convergence rate of the proposed algorithm can be easily obtained if the parameters $\alpha_{\max}$, $\tilde{\gamma}$ and Φ follow the conditions (36)–(38), respectively. It is noteworthy that these parameters only depend on the topology of the network and objective functions. Although some global parameters such as $\bar{\mu}$, $\bar{\psi}$ and $\tilde{\psi}$ are needed when designing step-sizes and the coefficients, these parameters can be easily pre-calculated without much effort.*

Remark 4. *Being uncoordinated and being nonidentical are two important characteristics often designed in many related studies, considering that step-sizes and coefficients might be changed with time variance in some practical scenarios. In our algorithm, step-sizes and coefficients were designed as uncoordinated, time-varying, and nonidentical. Furthermore, the largest step-size and coefficients were chosen according to their bounds shown in Theorem 1, which only depend on the the communication network and the objective functions. Notably, there is a bound of a conditional number, such that when the largest step-size is chosen, the smallest step-size needs to be chosen carefully.*

5. Numerical Experiments

In this section, some necessary numerical experiments in a real dataset are provided to illustrate the efficiency and superiority of our algorithm. In the experiments, we considered a binary-classification logistic-regression problem in the Wisconsin breast cancer dataset

provided in the UCI Machine Learning Repository [45]. The problem can be described in the following form:

$$\min_{x\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n}\frac{1}{n_i}\sum_{j=1}^{n_i}\ln\left(1+\exp\left(-y_{ij}a_{ij}^T x\right)\right)+\frac{\tau}{2}\|x\| \tag{50}$$

with each local objective function f_i written as follows:

$$f_i(x) = \frac{1}{n_i}\sum_{j=1}^{n_i}\ln\left(1+\exp\left(-y_{ij}a_{ij}^T x\right)\right)+\frac{\tau}{2}\|x\| \tag{51}$$

where n is the number of the agent in network, and d is the dimension of the decision variable. Each agent i is assumed to have an equal data samples n_i, i.e., $n_i = \frac{N}{n}$(N is the total data samples). $a_{ij} \in \mathbb{R}^d$ represents the feature vector of the jth data sample at the ith agent, while $y_{ij} \in \{-1, 1\}$ denotes the corresponding label. The regularization term $\frac{\tau}{2}\|x\|$ with parameter $\lambda = 1$ was set to avoid over-fitting.

In the experiments, we set $N = 200$ as training data, and $d = 9$ represents the feature in the real dataset. Meanwhile, we simulated a randomly undirected network generated by the Erdos–Renyi network with $n = 10$ nodes and edge probability $p = 0.7$. Then, we compared the proposed algorithm UGNH to relevant algorithms: EXTRA, HSADO, and UG.

Figures 1–5 show the results of our experiments, and the main conclusions are as follows:

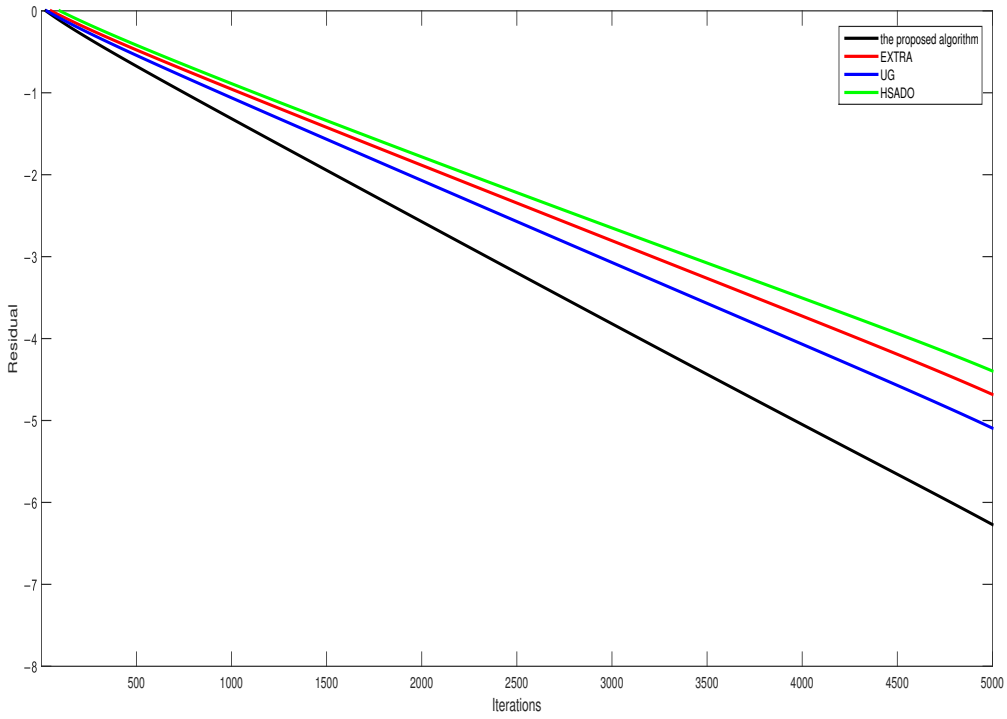

Figure 1. Performance comparisons between the proposed algorithm and related algorithms.

- Figure 1 indicates that the proposed algorithm UGNH promotes the convergence rate compared to the related algorithms in the real dataset; thus, UGNH is effective and superior. From Figure 2, the sequences generated by UGNH, EXTRA, UG, and HSADO can converge to the optimal solutions as expected. Avoiding confusion of the figure, only one dimension of each decision variable is exhibited.

- Figure 3 means that UGNH with the Nesterov momentum and the Heavy-ball momentum improved the convergence rate compared to the algorithm with only one or no momentum.
- In Figure 4, we can conclude that step-size is usually chosen very small; the larger step-size leads to a faster convergence rate if it is chosen under the upper bound. For the coefficient, a similar result can be obtained in Figure 5. Comparing the two figures, it can be concluded that small changes in step-size are more influential than that of the coefficient.

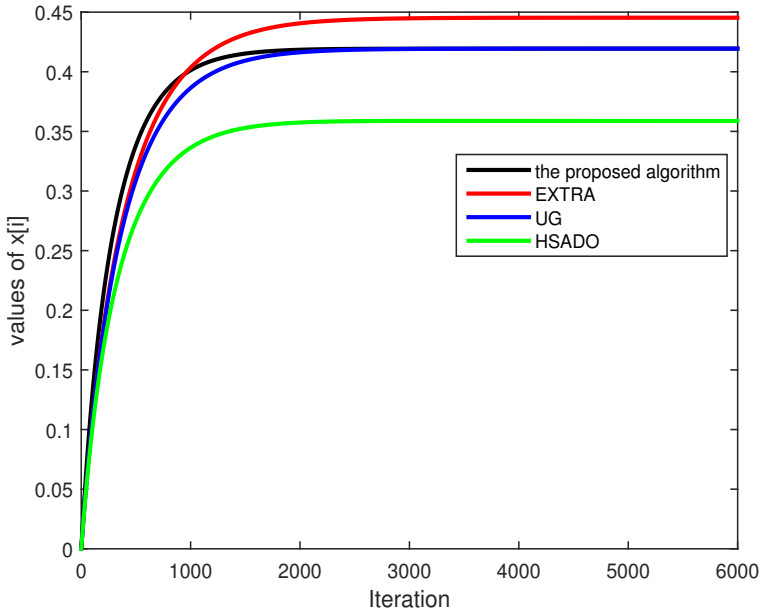

Figure 2. One dimension of variable between the proposed algorithm and related algorithms.

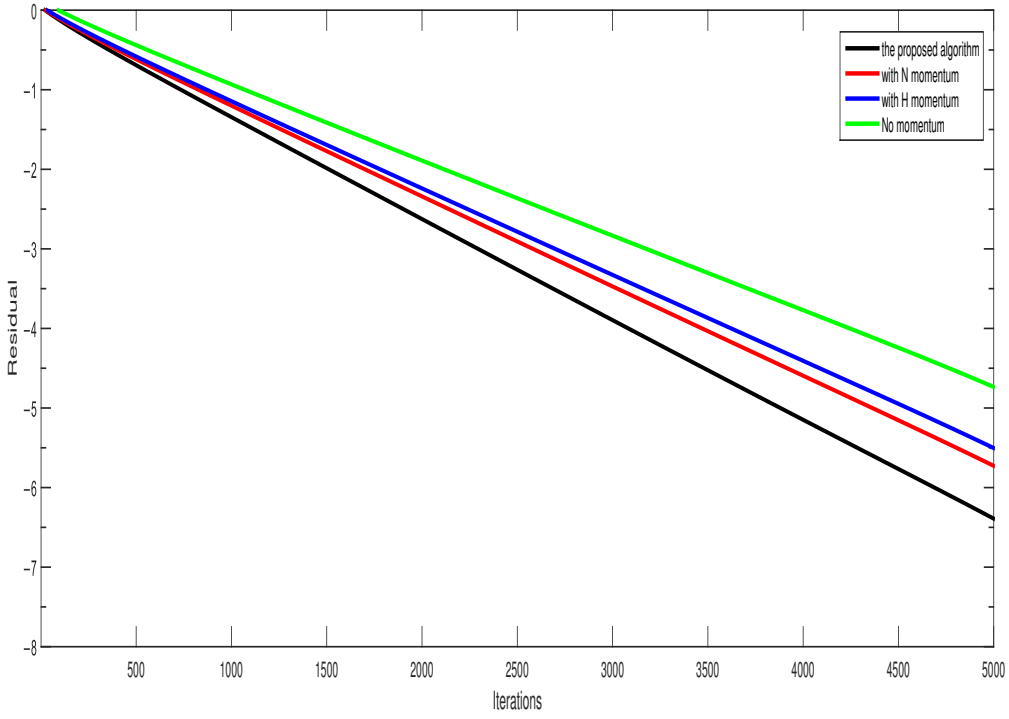

Figure 3. Performance comparisons between the proposed algorithm and the method without momentum terms.

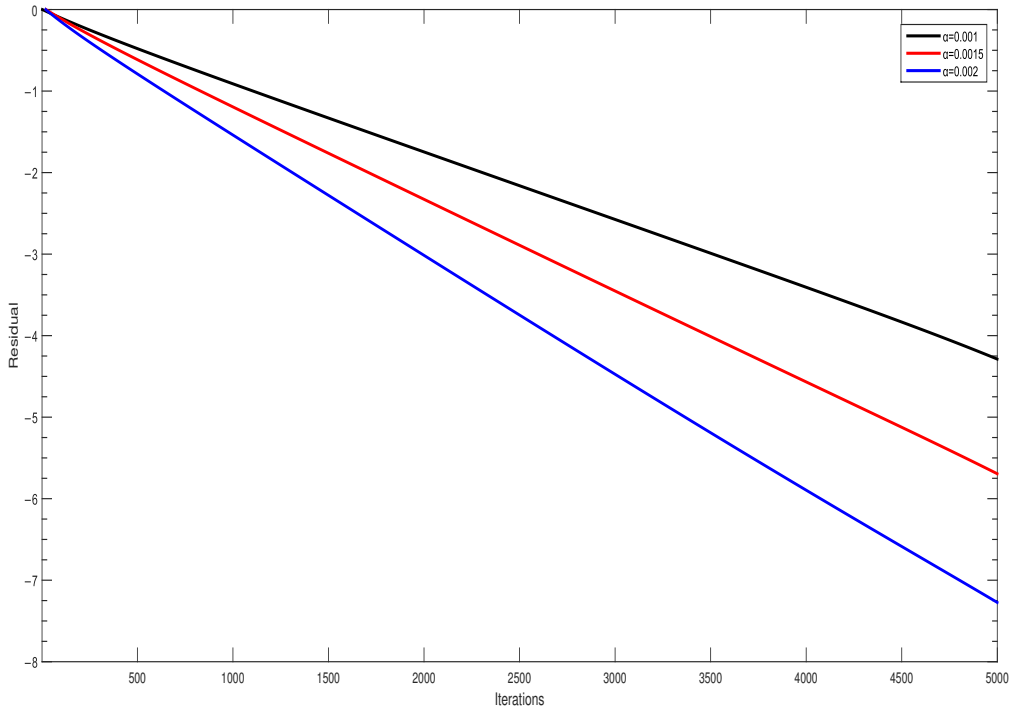

Figure 4. Performance comparisons between different step-sizes.

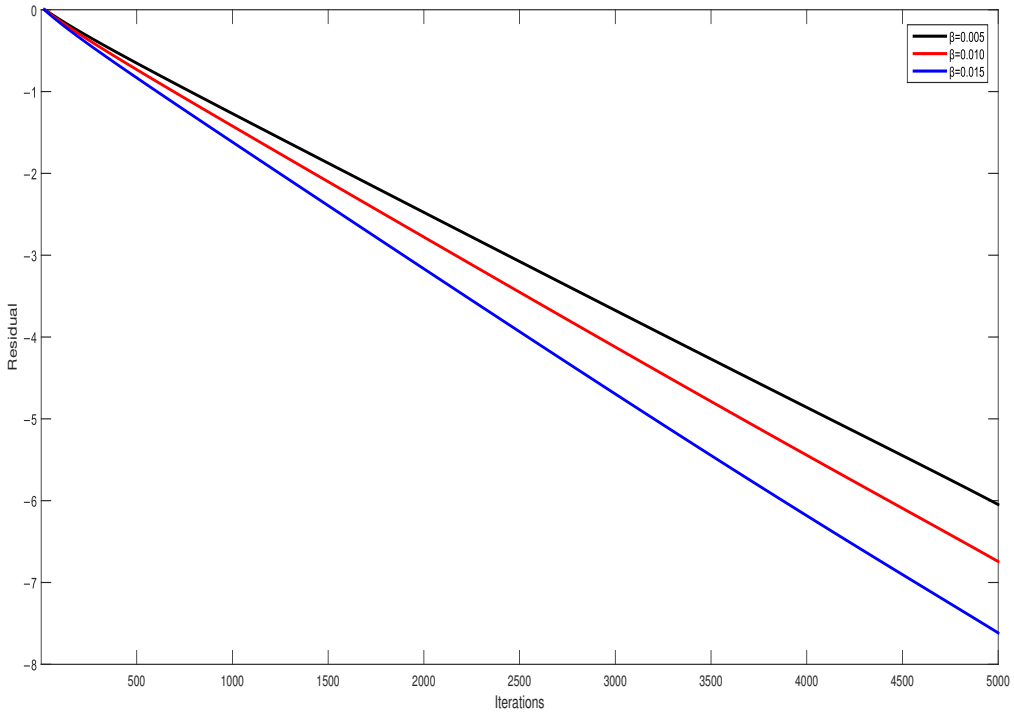

Figure 5. Performance comparisons between different momentum coefficients.

6. Conclusions

In this study, a novel uncoordinated, time-varying, and nonidentical distributed optimization accelerated algorithm was proposed. It was mainly applied to handle the distributed optimization convex problem in an undirected network, where all agents are in an effort to optimize the average of all local objective functions collaboratively. When the largest step-size and the maximum coefficient do not exceed some estimated upper bounds, which have been provided in Theorem 1, the convergence rate of UGNH is linear under the

condition that each local objective function is smooth and strongly convex. Besides, these parameters only depend on the topology of the network and the local objective function.

It is worth noting that to achieve a faster linear convergence rate, the Heavy-ball and Nesterov accelerated methods were simultaneously added into the algorithm, which provides a new way for accelerating convergence of other distributed optimization algorithms. Furthermore, the experiment results verified the effective and superior performance in a real dataset. However, UGNH is not suitable for all scenarios, and there are some more in-depth areas worth studying, such as the time-varying network architecture, random link failures, asynchronous communication between agents, directed networks, and so on. In all, these problems are worthy of further study and are our future research direction.

Author Contributions: Data curation, H.Z.; writing—original draft, Y.L.; supervision, X.G.; writing—review and editing, H.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the National Natural Science Foundation of China (41271292), in part by the Key Project of Chongqing Science and Technology Bureau (cstc2019jscx-gksbX0103), in part by the Fundamental Research Funds for the Central Universities under Project (SWU2009107), in part by the Chongqing Natural Science Foundation (cstc2020jcyj-msxmX0324), in part by the Key Project of Natural Science Research of Education Department in Anhui Province of China (KJ2019A0864), and in part by the Construction of Chengdu-Chongqing Economic Circle Science and Technology Innovation Project (KJCX2020007).

Institutional Review Board Statement: This paper does not studies involving human or animal.

Informed Consent Statement: This paper does not studies involving human or animal.

Data Availability Statement: The dataset can be fetched on the website http://archive.ics.uci.edu/ml, (accessed on 11 December 2021).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wang, H.; Liao, X.; Wang, Z.; Huang, T.; Chen, G. Distributed parameter estimation in unreliable sensor networks via broadcast gossip algorithms. *Neural Netw.* **2016**, *73*, 1–9. [CrossRef] [PubMed]
2. Dougherty, S.; Guay, M. An extremum-seeking controller for distributed optimization over sensor networks. *IEEE Trans. Autom. Control* **2016**, *62*, 928–933. [CrossRef]
3. Rahmani, A.M.; Ali, S.; Yousefpoor, M.S.; Yousefpoor, E.; Naqvi, R.A.; Siddique, K.; Hosseinzadeh, M. An area coverage scheme based on fuzzy logic and shuffled frog-leaping algorithm (sfla) in heterogeneous wireless sensor networks. *Mathematics* **2021**, *9*, 2251. [CrossRef]
4. Ren, W. Consensus based formation control strategies for multi-vehicle systems. In Proceedings of the 2006 American Control Conference, Philadelphia, PA, USA, 14–16 June 2006; p. 6.
5. Yan, B.; Shi, P.; Lim, C.C.; Wu, C.; Shi, Z. Optimally distributed formation control with obstacle avoidance for mixed-order multi-agent systems under switching topologies. *IET Control Theory Appl.* **2018**, *12*, 1853–1863. [CrossRef]
6. Cevher, V.; Becker, S.; Schmidt, M. Convex optimization for big data: Scalable, randomized, and parallel algorithms for big data analytics. *IEEE Signal Processing Mag.* **2014**, *31*, 32–43. [CrossRef]
7. Zhang, Z.; Wang, W.; Pan, G. A Distributed Quantum-Behaved Particle Swarm Optimization Using Opposition-Based Learning on Spark for Large-Scale Optimization Problem. *Mathematics* **2020**, *8*, 1860. [CrossRef]
8. Li, K.; Liu, Q.; Yang, S.; Cao, J.; Lu, G. Cooperative optimization of dual multiagent system for optimal resource allocation. *IEEE Trans. Syst. Man Cybern. Syst.* **2018**, *50*, 4676–4687. [CrossRef]
9. Jia, W.; Qin, S. Distributed Optimization Over Directed Graphs with Continuous-Time Algorithm. In Proceedings of the 2019 Chinese Control Conference (CCC), Guangzhou, China, 27–30 July 2019; pp. 1911–1916.
10. Ahmed, E.M.; Rathinam, R.; Dayalan, S.; Fernandez, G.S.; Ali, Z.M.; Aleem, S.H.; Omar, A.I. A Comprehensive Analysis of Demand Response Pricing Strategies in a Smart Grid Environment Using Particle Swarm Optimization and the Strawberry Optimization Algorithm. *Mathematics* **2021**, *9*, 2338. [CrossRef]
11. Zhang, Q.; Gong, Z.; Yang, Z.; Chen, Z. Distributed convex optimization for flocking of nonlinear multi-agent systems. *Int. J. Control Autom. Syst.* **2019**, *17*, 1177–1183. [CrossRef]
12. Tang, X.; Li, M.; Wei, S.; Ding, B. Event-triggered Synchronous Distributed Model Predictive Control for Multi-agent Systems. *Int. J. Control Autom. Syst.* **2021**, *19*, 1273–1282. [CrossRef]
13. Nedic, A.; Ozdaglar, A. Distributed subgradient methods for multi-agent optimization. *IEEE Trans. Autom. Control* **2009**, *54*, 48–61. [CrossRef]
14. DeGroot, M.H. Reaching a consensus. *J. Am. Stat. Assoc.* **1974**, *69*, 118–121. [CrossRef]

15. Ram, S.S.; Nedić, A.; Veeravalli, V.V. Distributed stochastic subgradient projection algorithms for convex optimization. *J. Optim. Theory Appl.* **2010**, *147*, 516–545.
16. Nedic, A.; Ozdaglar, A.; Parrilo, P.A. Constrained consensus and optimization in multi-agent networks. *IEEE Trans. Autom. Control* **2010**, *55*, 922–938. [CrossRef]
17. Duchi, J.C.; Agarwal, A.; Wainwright, M.J. Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Trans. Autom. Control* **2011**, *57*, 592–606. [CrossRef]
18. Jakovetić, D.; Xavier, J.; Moura, J.M. Fast distributed gradient methods. *IEEE Trans. Autom. Control* **2014**, *59*, 1131–1146. [CrossRef]
19. Shi, W.; Ling, Q.; Wu, G.; Yin, W. Extra: An exact first-order algorithm for decentralized consensus optimization. *SIAM J. Optim.* **2015**, *25*, 944–966. [CrossRef]
20. Shi, W.; Ling, Q.; Wu, G.; Yin, W. A proximal gradient algorithm for decentralized composite optimization. *IEEE Trans. Signal Processing* **2015**, *63*, 6013–6023. [CrossRef]
21. Xi, C.; Khan, U.A. DEXTRA: A fast algorithm for optimization over directed graphs. *IEEE Trans. Autom. Control* **2017**, *62*, 4980–4993. [CrossRef]
22. Zeng, J.; Yin, W. Extrapush for convex smooth decentralized optimization over directed networks. *arXiv* **2015**, arXiv:1511.02942.
23. Yuan, K.; Ying, B.; Zhao, X.; Sayed, A.H. Exact diffusion for distributed optimization and learning-Part I: Algorithm development. *IEEE Trans. Signal Processing* **2018**, *67*, 708–723. [CrossRef]
24. Yuan, K.; Ying, B.; Zhao, X.; Sayed, A.H. Exact diffusion for distributed optimization and learning-Part II: Convergence analysis. *IEEE Trans. Signal Processing* **2018**, *67*, 724–739. [CrossRef]
25. Jakovetić, D.; Moura, J.M.; Xavier, J. Linear convergence rate of a class of distributed augmented lagrangian algorithms. *IEEE Trans. Autom. Control* **2014**, *60*, 922–936. [CrossRef]
26. Qu, G.; Li, N. Harnessing smoothness to accelerate distributed optimization. *IEEE Trans. Control Netw. Syst.* **2017**, *5*, 1245–1260. [CrossRef]
27. Nedic, A.; Olshevsky, A.; Shi, W. Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM J. Optim.* **2017**, *27*, 2597–2633. [CrossRef]
28. Jakovetic, D.; Krejic, N.; Malaspina, G. Linear Convergence Rate Analysis of a Class of Exact First-Order Distributed Methods for Time-Varying Directed Networks and Uncoordinated Step Sizes. *arXiv* **2007**, arXiv:2007.08837 2020.
29. Nedić, A.; Olshevsky, A.; Shi, W.; Uribe, C.A. Geometrically convergent distributed optimization with uncoordinated step-sizes. In Proceedings of the 2017 American Control Conference (ACC), Seattle, WA, USA, 24–26 May 2017; pp. 3950–3955.
30. L′u, Q.; Li, H.; Xia, D. Geometrical convergence rate for distributed optimization with time-varying directed graphs and uncoordinated step-sizes. *Inf. Sci.* **2018**, *422*, 516–530. [CrossRef]
31. Qu, G.; Li, N. Accelerated distributed Nesterov gradient descent. *IEEE Trans. Autom. Control* **2019**, *65*, 2566–2581. [CrossRef]
32. Xin, R.; Khan, U.A. Distributed heavy-ball: A generalization and acceleration of first-order methods with gradient tracking. *IEEE Trans. Autom. Control* **2019**, *65*, 2627–2633. [CrossRef]
33. Mokhtari, A.; Ribeiro, A. DSA: Decentralized double stochastic averaging gradient algorithm. *J. Mach. Learn. Res.* **2016**, *17*, 2165–2199.
34. Nedić, A.; Ozdaglar, A. Subgradient methods for saddle-point problems. *J. Optim. Theory Appl.* **2009**, *142*, 205–228. [CrossRef]
35. Jakovetić, D. A unification and generalization of exact distributed first-order methods. *IEEE Trans. Signal Inf. Processing Over Netw.* **2018**, *5*, 31–46. [CrossRef]
36. Xu, J.; Zhu, S.; Soh, Y.C.; Xie, L. Augmented distributed gradient methods for multi-agent optimization under uncoordinated constant stepsizes. In Proceedings of the 2015 54th IEEE Conference on Decision and Control (CDC), Osaka, Japan, 15–18 December 2015; pp. 2055–2060.
37. Li, H.; Zheng, Z.; Lü, Q.; Wang, Z.; Gao, L.; Wu, G.C.; Ji, L.; Wang, H. Primal-Dual Fixed Point Algorithms Based on Adapted Metric for Distributed Optimization. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *2021*, 1–15. [CrossRef]
38. Liu, P.; Li, H.; Dai, X.; Han, Q. Distributed primal-dual optimisation method with uncoordinated time-varying step-sizes. *Int. J. Syst. Sci.* **2018**, *49*, 1256–1272. [CrossRef]
39. Nesterov, Y. *Introductory Lectures on Convex Optimization: A Basic Course*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2003; Volume 87.
40. Rivet, A.; Souloumiac, A. *Introduction to Optimization. Optimization Software, Publications Division*; Citeseer: Washington, DC, USA, 1987.
41. Xin, R.; Khan, U.A. A linear algorithm for optimization over directed graphs with geometric convergence. *IEEE Control Syst. Lett.* **2018**, *2*, 315–320. [CrossRef]
42. Cheng, H.; Li, H.; Wang, Z. On the convergence of exact distributed generalisation and acceleration algorithm for convex optimisation. *Int. J. Syst. Sci.* **2020**, *51*, 1–17. [CrossRef]
43. Lü, Q.; Liao, X.; Li, H.; Huang, T. A nesterov-like gradient tracking algorithm for distributed optimization over directed networks. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, *51*, 6258–6270 [CrossRef]
44. Hestenes, M.R.; Stiefel, E. *Methods of Conjugate Gradients for Solving Linear Systems*; NBS: Washington, DC, USA, 1952; Volume 49.
45. Dua, D.; Graff, C. UCI Machine Learning Repository. 2017. Available online: http://archive.ics.uci.edu/ml (accessed on 11 December 2021).