Scilight

Journal of Machine Learning and Information Security
https://www.sciltp.com/journals/jmlis



Check for updates

Review

Comprehensive Survey of Distributed Optimization: Challenges, Strategies, and Applications

Bingxue Luo

Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, 20133 Milan, Italy; papayavein@gmail.com

How To Cite: Luo, B. Comprehensive Survey of Distributed Optimization: Challenges, Strategies, and Applications. *Journal of Machine Learning and Information Security* **2026**, *2*(4), 21. https://doi.org/10.53941/jmlis.2026.100021

Received: 18 June 2026
Revised: 20 September 2026
Accepted: 30 September 2026
Published: 20 October 2026

Abstract: Distributed optimization plays an important role in multi-node collaborative computing and has developed rapidly in recent years, both in theory and applications. However, most classical distributed optimization algorithms are generally developed under idealized communication settings, which, to some extent, limit their applicability to practical systems with complex constraints. This survey aims to bridge the gap between theory and practice in distributed optimization by examining its theoretical foundations, recent progress, and major open challenges. Initially, we outline some theoretical breakthroughs regarding graph topologies, objective functions, and acceleration strategies. The discussion then turns to approaches for addressing practical constraints like stochastic noise, limited bandwidth, and asynchronous delays. Finally, we present several directions that are worth exploring in the future.

Keywords: multi-node system; distributed algorithms; stochastic optimization; acceleration methods; communication strategies; asynchronous update

1. Introduction

Large-scale optimization problems now arise frequently in engineering and scientific applications, especially with the growing use of big data and artificial intelligence [1]. In conventional centralized architecture, data are usually gathered at a central node for processing. This design, however, becomes difficult to maintain when the amount of data and the size of the network increase. As a single node may not have sufficient storage or computational resources to handle large datasets efficiently, and collecting data from all nodes can generate considerable communication traffic. Centralized systems are also sensitive to single-point failures and often scale poorly as the network grows. More importantly, in many applications, privacy and security concerns often prevent raw data from being directly shared across nodes. These limitations have encouraged the development of alternative optimization paradigms that can better support large-scale and networked applications.

Distributed optimization offers a way to overcome the limitations of centralized computing. In this setting, multiple nodes are connected through a communication network and work together without relying on a global central coordinator. Each node is only permitted to access its local data and cost function. The network can therefore optimize the global objective function by performing local computations and exchanging information with neighboring nodes [2]. As opposed to traditional centralized strategies, distributed optimization is able to improve scalability and resilience by distributing computation and storage loads across nodes.

Leveraging their superior characteristics, distributed algorithms have attracted growing attention from both academia and industry in recent years. Prominent application areas for these algorithms include, but are not limited to, cooperative control [3], sensor networks [4], and constrained optimization [5]. Many foundational works have been proposed during the early stages of this field, including the classical distributed gradient descent (DGD) [6] method and its enhanced version EXTRA [7]. By combining local gradient updates with consensus averaging among neighboring nodes, these methods provide an important theoretical basis for the subsequent development of distributed optimization. However, the convergence performance of these algorithms often relies on idealized assumptions, such as synchronous settings, strongly convex objective functions, and undirected communication graphs.



Copyright: © 2026 by the authors. This is an open access article under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).
Publisher's Note: Scilight stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

To relax the aforementioned conditions and broaden the applicability of distributed algorithms, subsequent studies have made considerable progress in several directions. For network topology, communication networks are no longer restricted to undirected graphs, but have gradually been extended to directed graphs [8] and highly dynamic time-varying graphs [9]. The range of objective functions has expanded to composite optimization with non-differentiable terms [10], non-convex problems [11], and time-varying optimization in which local objectives and optimizers may change over time [12]. More general dynamical models of the participating nodes have also been considered in [13]. Furthermore, by integrating momentum mechanisms [14] and proximal bundle methods [15], related algorithms can achieve faster convergence under suitable assumptions. Practical implementations must also account for limited information and system resources. Stochastic algorithms [16] can work with noisy gradients, while communication-efficient methods reduce the communication burden between nodes [17]. In addition, for networks where updates cannot be perfectly synchronized, asynchronous algorithms [18] provide greater flexibility in handling delays.

The literature already contains several surveys on distributed optimization [19–22], each emphasizing a different aspect of the topic. Some studies concentrate on the evolution of network topology in discrete- or continuous-time settings [19], while others emphasize particular convex optimization [20] or specific problem classes [21, 22]. Our survey takes a broader view by discussing theoretical progress together with the practical challenges that affect the design and implementation of distributed algorithms.

The rest part of this survey is structured as: Section 2 first introduces the background required for the subsequent analysis, including graph theory, network models, and several useful inequalities. Paved by these knowledge, Section 3 reviews the theoretical development of distributed optimization in terms of network topology, objective functions, and acceleration strategies. Section 4 shifts the focus to practical settings and discusses stochastic optimization, communication efficiency, and asynchronous computation. The main conclusions and several open research directions are presented in Section 5.

2. Preliminaries

We first present the network model and formulate the distributed optimization problem. Several useful inequalities for the subsequent convergence analysis are also summarized.

2.1. Graph Theory and Network Models

Communication and information exchange between nodes are often modeled with graph theory in distributed optimization. Generally, it can be represented by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, 2, \ldots, n\}$ denotes a set of n nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the edge set. A pair $(j, i) \in \mathcal{E}$ indicates that information can be transmitted from node j to node i. The graph then may take different forms depending on the direction of information exchange and whether communication links vary over time.

The undirected graph is commonly used to model networks with bidirectional, symmetric communication links. Within this topology, a communication link between nodes i and j dictates that both $(i, j) \in \mathcal{E}$ and $(j, i) \in \mathcal{E}$, enabling two-way information exchange. Additionally, such a graph is considered connected if a path exists between any two distinct nodes in the network. And the neighbor set of node i is $\mathcal{N}^i = \{j | (j, i) \in \mathcal{E}, j \neq i\}$. To facilitate information fusion, a non-negative mixing weight matrix $W = [w_{ij}] \in \mathbb{R}^{n \times n}$ is assigned to the system, where $w_{ij} > 0$ if node $j \in \mathcal{N}^i$, otherwise $w_{ij} = 0$. For a connected undirected graph, its weight matrix W is commonly assumed to satisfy the following properties:

$$W = W^T, \quad W\mathbf{1}_n = \mathbf{1}_n, \quad \mathbf{1}_n^T W = \mathbf{1}_n^T, \tag{1}$$

where $\mathbf{1}_n$ represents an n-dimensional column vector of all ones. Therefore, an undirected graph is inherently balanced because its bidirectional links guarantee that every node's in-degree equals its out-degree. These properties allow information to be exchanged symmetrically among neighboring nodes and are widely used in consensus-based distributed optimization.

However, the communication among nodes is typically unidirectional since there are many differences in broadcasting power and asymmetries in channel bandwidth. In this case, the network needs to be modeled as a directed graph $\mathbb{G} = (\mathcal{V}, \mathcal{E})$. The in-neighbor and out-neighbor sets of node i can therefore be defined as follows according to the flow of information:

$$\mathcal{N}_{in}^i = \{j | (j, i) \in \mathcal{E}\}, \quad \mathcal{N}_{out}^i = \{j | (i, j) \in \mathcal{E}\}. \tag{2}$$

Furthermore, for any two distinct nodes in $\mathbb{G}$, if there exists a directed path between them, then $\mathbb{G}$ is said to be

strongly connected. Naturally, the inherent structure of $\mathbb{G}$ makes it impossible to construct a doubly stochastic matrix like the undirected graph $\mathcal{G}$ does. Consequently, for the algorithm over directed networks, its associated weight matrix typically turns into either a row-stochastic matrix $A = [a_{ij}]$ or a column-stochastic matrix $B = [b_{ij}]$:

$$A\mathbf{1}_n = \mathbf{1}_n, \quad \mathbf{1}_n^T B = \mathbf{1}_n^T. \tag{3}$$

In dynamic communication environments, network topologies often change over time, which calls for the introduction of time-varying graphs $\mathcal{G}_k = (\mathcal{V}, \mathcal{E}_k)(\mathbb{G}_k = (\mathcal{V}, \mathcal{E}_k))$. Accordingly, the neighbor set $\mathcal{N}^{i,k}$ ($\mathcal{N}_{in}^{i,k}$, $\mathcal{N}_{out}^{i,k}$) of node i and the weight matrix W_k (A_k, B_k) are replaced by their time-varying variables.

2.2. Mathematical Formulation of Distributed Optimization

The primary objective of multi-node collaborative optimization in distributed networks is to minimize a global cost function. And this objective is conventionally taken as the sum or average of the local cost functions across all nodes, which can be expressed as:

$$\min_{x \in \mathbb{R}^p} f(x) = \frac{1}{n} \sum_{i=1}^{n} f^i(x), \tag{4}$$

where $x \in \mathbb{R}^p$ is decision variable, n denotes the number of nodes, and $f^i : \mathbb{R}^p \to \mathbb{R}$ signifies the local cost function exclusively known to node i. To compute in a distributed way, we let each node maintain a local copy $x^i \in \mathbb{R}^p$ of the decision variable. Hence, the primal problem can be restructured as:

$$\min_{x^1,\ldots,x^n} \sum_{i=1}^{n} f^i(x^i), \text{ s.t. } x^i = x^j, \ \forall (i,j) \in \mathcal{E}. \tag{5}$$

The above equality constraint ensures that, as the algorithm iterates, the local variable copies of all nodes will ultimately reach consensus, i.e., $x^1 = x^2 = \cdots = x^n$. This ensures that the solution obtained through distributed computation is consistent with the global optimum.

Machine learning tasks involving constraints often feature more complex objective functions, which means the problem (5) can be further extended into the following composite optimization model:

$$\min_{x^1,\ldots,x^n} \sum_{i=1}^{n} (f^i(x^i) + h^i(x^i)), \text{ s.t. } x^i = x^j, \ \forall (i,j) \in \mathcal{E}. \tag{6}$$

where $f^i(x^i)$ is generally a smooth and differentiable function used to characterize the data fidelity term, whereas $h^i(x^i)$ represents a non-smooth and non-differentiable function acting as a regularization penalty term. Common examples for $h^i(x)$ include the ℓ_1-norm designed to induce sparsity, or an indicator function defining a feasible constraint set.

2.3. Core Inequalities for Analysis

To establish strict convergence guarantees and derive convergence rates within the theoretical analysis of distributed optimization, it is generally required to impose certain assumptions on the local cost functions. A prevalent assumption in the analysis of most first-order distributed algorithms is that the local objective $f^i(x)$ is smooth, that is, its gradient exhibits Lipschitz continuity. Specifically, it is assumed that there exists a constant $L > 0$ such that for any $x, y \in \mathbb{R}^p$, the following inequality holds:

$$||\nabla f^i(x) - \nabla f^i(y)|| \leq L||x - y||. \tag{7}$$

Based on this assumption, we can derive a key property of smooth functions, which will be essential for proving the descent of the algorithm in each iteration:

$$f^i(y) \leq f^i(x) + \nabla f^i(x)^T (y - x) + \frac{L}{2}||y - x||^2. \tag{8}$$

Convexity of the objective function serves as a theoretical basis that ensures the convergence of distributed algorithms to a global optimal solution. If there exists a convex function $f^i(x)$, then it always lies above its tangent at any point, that is

$$f^i(y) \geq f^i(x) + \nabla f^i(x)^T (y - x), \tag{9}$$

To obtain a faster convergence rate, such as linear convergence, the objective function is often further assumed to be μ-strongly convex ($\mu > 0$). This demands that the function is more than simply convex, but is strictly bounded below by a quadratic function:

$$f^i(y) \geq f^i(x) + \nabla f^i(x)^T(y-x) + \frac{\mu}{2}||y-x||^2. \tag{10}$$

Owing to strong convexity, we are assured of both the existence and uniqueness of the global optimal solution [23].

As for non-convex settings, distributed algorithms can usually only converge to first-order stationary points (i.e., $\nabla f(x) = 0$) since there is no global lower-bound structure provided by convexity. To achieve global convergence in nonconvex environments, we introduce the Polyak–Łojasiewicz (PŁ) condition into the analysis. Specifically, when the global objective function $f(x)$ satisfies the P-Ł condition, there exists a constant $\mu > 0$ such that:

$$\frac{1}{2}||\nabla f(x)||^2 \geq \mu(f(x) - f^*), \ \forall x \in \mathbb{R}^p, \tag{11}$$

where f^* is the global minimum function value.

3. Algorithm Evolution from Theoretical Perspectives

This section systematically reviews the evolution of distributed algorithms from network topology, objective functions, and convergence efficiency.

3.1. Algorithm Design over Complex Graph Topologies

The communication network among nodes is important in shaping how information is exchanged in distributed optimization. As application scenarios become more complex, the design of communication topology for distributed algorithms has evolved from undirected to directed graphs and further to time-varying graphs.

Early distributed optimization algorithms are typically developed under undirected network topologies, where doubly stochastic weight matrices can be constructed. For example, the local update rule of the DGD [6] algorithm is as follows:

$$x_{k+1}^i = \sum_{j \in \mathcal{N}_i} w_{ij} x_k^i - \alpha_k \nabla f^i(x_k^i), \tag{12}$$

where x_k^i is the local estimate of node i at the k-th iteration, w_{ij} is the doubly stochastic mixing weight, and α_k is the step size. To eliminate the steady-state error caused by a fixed step size, DGD generally employs a diminishing step size, resulting in a sublinear convergence rate of $\mathcal{O}(\ln k/\sqrt{k})$ or $\mathcal{O}(\ln k/k)$. Building upon the DGD algorithm, EXTRA [7] incorporates a historical gradient correction term to achieve exact convergence under a fixed step size. Its iteration rule is formulated as:

$$x_{k+2}^i = x_{k+1}^i + \sum_{j=1}^n w_{ij} x_{k+1}^j - \sum_{j=1}^n \tilde{w}_{ij} x_k^j - \alpha[\nabla f^i(x_{k+1}^i - x_k^i)], \tag{13}$$

and $\tilde{W} = [\tilde{w}_{ij}]$ is another symmetric mixing matrix (typically chosen as $\tilde{W} = \frac{I+W}{2}$). Under strongly convex conditions, EXTRA can achieve a specific convergence rate $\mathcal{O}(1/k)$ or even linear convergence. However, it strictly relies on the symmetry of the weight matrix, which means the underlying communication topology is always an undirected graph.

Communication between nodes may be asymmetric in practical networks, with information flowing only in one direction along some links. Such networks are therefore described by directed graphs, where a doubly stochastic mixing matrix is generally unavailable. One common approach for this structure is to use a column-stochastic matrix together with push-sum protocol [24]. Algorithms such as DEXTRA [25], ADD-OPT [26], and Push-DIGing [27] follow this idea. These algorithms introduce an auxiliary variable to asymptotically track the right eigenvector of the column-stochastic matrix. A practical limitation of column-stochastic mixing is that each node generally needs to know its out-degree, which is not always available in broadcast-based communication. Row-stochastic alternatives were therefore developed, with FROST [28] being a representative example. A key benefit of row-stochastic matrices is their ease of implementation, as each node can simply assign weights locally based on the information it receives, without requiring any knowledge of the out-degree information. Unfortunately, both classes of algorithms require the learning of eigenvectors, increasing computational and communication costs. In response to this, the Push-Pull [29] and $\mathcal{AB}$ [30] algorithms were developed. These two techniques use a row-stochastic matrix $A = \{a_{ij}\}$ and a column-stochastic

matrix $B = \{b_{ij}\}$ simultaneously to pull decision variables and push gradient information:

$$x_{k+1}^i = \sum_{j=1}^{n} a_{ij} x_k^j - \alpha y_k^i, \tag{14}$$

$$y_{k+1}^i = \sum_{j=1}^{n} b_{ij} y_k^j + \nabla f^i(x_{k+1}^i) - \nabla f^i(x_k^i), \tag{15}$$

where y_k^i serves as an auxiliary variable for tracking the global average gradient. Notably, extensions of the above methods to dynamic environments with time-varying graphs have been proposed in [31,32].

3.2. Evolution and Extension of Objective Functions

The assumptions for objective functions in distributed optimization have experienced two critical advancements. Their applicability was first generalized from smooth differentiable functions to composite optimization with non-differentiable terms, and later, motivated by practical machine learning needs, was further extended from convex to non-convex settings.

Most conventional DGD algorithms [6,7] were developed for smooth and differentiable objective functions. Yet in modern machine learning tasks, composite optimization problems of the form (6) are commonly encountered. For such problems, DDA [33] usually updates its dual variables by combining the weighted average with the local subgradient and then drives the new primal iterates via a proximal projection. While effective in suppressing consensus error, the use of diminishing step sizes in DDA inevitably sacrifices convergence speed. Driven by the need to achieve exact convergence with a constant step size, the DADMM [34] algorithm was introduced. Building on this idea, subsequent studies have led to the PG-ADMM [35] algorithm. We note, however, that such exact methods become computationally expensive when applied to complex cost functions or high-dimensional data. IC-ADMM and IDC-ADMM [36] replace the exact ADMM step with an inexact one, reducing the per-iteration computational burden while maintaining convergence. In PG-EXTRA [37], the smooth part is updated through a local gradient step, while the non-smooth part is resolved through a proximal operator. Utilizing a fixed step size, it attains an $\mathcal{O}(1/k)$ ergodic convergence rate over the averaged iterate sequence.

Recent work has extended distributed optimization to nonconvex problems motivated by nonlinear models. In particular, NEXT [38] avoids directly optimizing the local non-convex objective by introducing a strongly convex surrogate at each node. A dynamic consensus scheme is then used to track the global gradient, leading to convergence toward a stationary solution. For composite non-convex problems, Prox-DGD [39] with a fixed step size generally converges only to a neighborhood of the stationary point. Furthermore, PG-ExtraPush [40] extends PG-EXTRA to directed networks by incorporating the push-sum protocol, retaining comparable convergence guarantees for convex objectives. Another extension is SONATA [41], which uses successive convex approximation (SCA) to address non-smooth composite non-convex optimization over time-varying directed graphs. More recently, ProxGT [42] provided a unified framework for several problem classes and established sample complexity bounds that do not depend on the network topology.

3.3. Acceleration Mechanisms for Enhanced Convergence

Slow convergence is a common limitation of first-order methods, which has led to extensive work on acceleration techniques. Among them, Heavy-Ball (HB) [43] and Nesterov's Accelerated Gradient (NAG) [44] are two widely used momentum-based acceleration methods. Both make use of information from previous iterations to modify the current update direction. Specifically, HB introduces a momentum term derived from the preceding two iterations:

$$v_{k+1}^i = v_k^i - \alpha \nabla f(v_k^i) + \beta(v_k^i - v_{k-1}^i), \tag{16}$$

where α is the step size, and β is the momentum parameter. By contrast, NAG evaluates the gradient at a lookahead point, which yields an accelerated rate of $\mathcal{O}(1/k^2)$ (compared with the $\mathcal{O}(1/k)$ rate of standard gradient descent) in certain convex settings. This operation involves two steps:

$$y_{k+1}^i = x_k^i - \alpha \nabla f(x_k^i), \tag{17}$$

$$x_{k+1}^i = y_{k+1}^i + \beta(y_{k+1}^i - y_k^i). \tag{18}$$

Several distributed algorithms have been developed by incorporating these two momentum-based mechanisms into the update process. With the Nesterov momentum term, Acc-DNGD [45] can achieve linear for smooth and

strongly convex functions. Under similar conditions, the work in [46] also established linear convergence by introducing the HB method into a proportional-integral consensus framework. $\mathcal{AB}m$ [47] and $\mathcal{AB}\mathcal{N}$ [48] fuse HB and NAG to improve convergence rate, respectively. The combination of HB and NAG can lead to promising results in recent studies. For instance, NHDA [49] extended FROST [28] by incorporating both HB and NAG, while DSUA-TV [50] combined distributed stochastic gradient tracking (DSGT) [51], DSGT-HB [52], and DSGT-NAG [52] through the use of a flexible interpolation factor. Similar acceleration methods can be found in [53,54]. However, most momentum-based methods are limited by their reliance on fixed parameters. Adaptive schemes address this issue by allowing the acceleration parameters to vary during the optimization process. By introducing the exponential moving average (EMA) for adaptive estimation, DADAM [55] provided an improved acceleration performance. Following this, GTAdam [56] successfully mitigated the challenges of data heterogeneity through the integration of gradient tracking. Another acceleration strategy is to exploit historical objective and gradient information. More specifically, DBM [57] significantly enhanced the convergence speed by constructing a piecewise linear approximation of the local cost function. Additionally, DPBM [58] extended Prox-DGD [39] by introducing a proximal bundle mechanism, with convergence established under a fixed step size.

Acceleration techniques have also been applied to practical energy-management problems. Leveraging second-order information, the research in [59] designed a distributed accelerated algorithm for integrated energy systems to improve convergence efficiency. Another strategy focuses on directly controlling the time required to reach the optimizer. focuses on directly controlling the time required to reach the optimizer. Fixed-time distributed optimization guarantees convergence within a finite upper bound that is independent of the initial states. For example, the study in [60] developed a distributed fixed-time algorithm for multi-agent systems with strongly convex local cost functions, and it was later extended by introducing coupled objective functions [61]. Prescribed-time optimization further strengthens this property by allowing the convergence time to be specified in advance. To this end, a two-stage prescribed-time distributed scheme was proposed in [62]. In addition, by considering both constrained and unconstrained problems, the work in [63] guaranteed convergence within an arbitrarily prescribed time. These approaches provide a different perspective on acceleration from gradient-based methods, since their main objective is to offer explicit control over the settling time rather than only improving the asymptotic convergence rate. This feature can be particularly useful for distributed systems with strict time requirements.

4. Frontier Challenges under Practical Constraints

When practical system constraints are taken into account, distributed optimization often faces additional challenges. This section discusses three representative aspects of these challenges, covering stochastic optimization, communication strategies, and asynchronous updates.

4.1. Stochastic Distributed Optimization

Evaluating the full gradient at each iteration is rarely feasible for large-scale machine learning tasks. As an alternative, most distributed stochastic algorithms rely on the stochastic first-order oracle ($\mathcal{SFO}$) to obtain a stochastic gradient $g^i(x_k^i, \xi_k^i)$, which is commonly assumed to satisfy

$$\text{(a)} \quad \mathbb{E}[g^i(x_k^i, \xi_k^i)|\mathcal{F}_k] = \nabla f^i(x_k^i), \tag{19}$$

$$\text{(b)} \quad \mathbb{E}[||g^i(x_k^i, \xi_k^i) - \nabla f^i(x_k^i)||^2|\mathcal{F}_k] \leq \sigma^2, \tag{20}$$

where the first equality ensures that the stochastic gradient is unbiased, and the second bounds its variance by a non-negative constant. Under these assumptions, DSGD [64] avoids repeated full-gradient evaluations and provides a practical approach to large-scale distributed stochastic optimization. However, it may converge slowly when the data are highly heterogeneous. Gradient-tracking variants of DSGD, such as DSGT [51] and GT-DSGD [65], were later developed to address this problem. These two methods can mitigate the impact of data heterogeneity by continuously tracking the global average stochastic gradient. DSGT-HB [52] and DSGT-NAG [52] further extend DSGT by introducing the Heavy-Ball method and Nesterov momentum term, respectively. For time-varying directed networks, Push-ASGD [66] incorporates Heavy-Ball into the gradient-push framework to tackle distributed non-convex stochastic problems.

The aforementioned algorithms have successfully reduced the overall computational costs. Yet, a fundamental issue persists: the inherent variance of stochastic gradients hinders their exact convergence to the global optimal solution. This has motivated researchers to propose variance reduction techniques, such as SAGA [67] and SVRG [68], to achieve a trade-off between computational cost and convergence accuracy. Through gradient tracking, GT-VR [69] effectively unifies these two techniques. In addition, a special case of GT-VR, GT-SAGA [70] has been proven to

converge to a first-order stationary solution for non-convex problems. For composite optimization problems, the work in [71] combines SAGA with stochastic gradient projection, resulting in a computationally efficient distributed algorithm. Compared with SAGA, distributed algorithms built on SVRG have received relatively less attention. It is therefore worthwhile to develop more distributed SVRG variants and to seek faster convergence while preserving the reduced computational cost of stochastic optimization.

4.2. Communication Efficient Strategies

Frequent information exchange among nodes incurs substantial communication overhead as network and data scales grow. Communication-efficient methods are therefore important for reducing this burden in practical distributed systems. Time-triggered communication was used in early distributed algorithms [72, 73], where information was sampled and exchanged periodically according to a predefined schedule. Nevertheless, this periodic communication may lead to unnecessary computational overhead. Event-triggered communication [74] offers a more flexible way to periodic updates. Rather than transmitting information at fixed intervals, a node communicates only when its measurement error exceeds a predefined threshold. This can substantially reduce redundant exchanges without affecting convergence. Specifically, by combining the event-triggered mechanism and Heavy-ball method, the work in [75] achieved a balance between communication efficiency and accelerated convergence. Under the PL condition, the method in [76] established global optimality while substantially reducing communication frequency. Recent advances have also explored event-triggered mechanisms for prescribed-time distributed optimization. For instance, an event-triggered control was introduced to conserve system resources in [62]. This idea was further extended to practical applications in [77], where a dynamic event-triggered scheme was applied to multienergy economic dispatch.

Another promising approach to reducing communication overhead is communication compression. Nodes may compress the exchanged information through quantization [78] or sparsification [79], avoiding the need to transmit full high-dimensional vectors. This can be observed in QSGD [80], which utilizes unbiased quantizers to encode and transmit gradient information efficiently. However, directly applying compression in distributed optimization may cause algorithm divergence due to the accumulation of errors over iterations. To mitigate this effect, the error compensation technique has been proposed. Among them, CHOCO-SGD [81] provided a representative example of integrating error compensation into distributed stochastic optimization. As an extension of DADAM [34], QPS-DADAM [82] integrated a random quantization scheme to mitigate communication overhead over directed networks. Breakthroughs have also been made in communication compression for distributed non-convex optimization. By integrating a compression mechanism, the work in [83] achieved sublinear convergence to a first-order stationary point. Moreover, DaSHCo [84] further combined communication compression with momentum acceleration, delivering significant improvements in communication efficiency while maintaining fast convergence. Despite these advances, communication efficiency still needs to be balanced against convergence speed and accuracy. Future research may explore communication compression together with mechanisms that provide stronger convergence performance, such as prescribed-time convergence guarantees.

4.3. Asynchronous and Delayed Updates

While the previously discussed methods have achieved significant performance from multiple aspects, the requirement of synchronization among nodes continues to limit the efficiency of distributed systems [85]. Synchronous algorithms, such as EXTRA [7], ADD-OPT [26], and Push-DIGing [27], are inevitably affected by the straggler effect. Instead of waiting for synchronized updates, asynchronous algorithms allow nodes to update independently based on the most recently available neighbor information. Such behavior avoids the need for a global clock and reduces the waiting time caused by slow nodes or communication links.

Many classical distributed methods have been extended to operate asynchronously. In asynchronous ADMM [86], nodes updated independently without global coordination. However, its performance is sensitive to stale information and may degrade under unbounded communication delays. By adopting fixed step sizes that do not depend on the communication delays, asynchronous DGD and DGD-ATC [87] avoided this problem and achieved exact convergence. Under asynchronous updates, ASY-SONATA [88] also established an exact linear convergence guarantee for strongly convex problems over directed networks. Furthermore, the study in [89] integrated projection operations with adaptive updates to tackle constrained optimization in asynchronous settings. Asynchronous methods have also been developed for composite problems with nonsmooth terms. For instance, AD-PDFBS [90] removed the need for global synchronization when solving composite convex problems subject to equality constraints and provided a strict convergence guarantee simultaneously. R-TriPD-Dist [91] further extended asynchronous updates to nonsmooth composite problems through a distributed primal-dual proximal scheme. For

composite nonconvex optimization, the work in [92] developed an asynchronous gradient tracking mechanism and derived a sublinear convergence rate under more general assumptions. Note that many existing analyses still rely on bounded-delay assumptions or specific activation models, while larger delays may require conservative step sizes and slow down convergence. It is worth exploring asynchronous algorithms that can handle more delay patterns without losing large step sizes or fast convergence.

5. Conclusions and Future Work

This survey provides an overview of the main developments in distributed optimization. Overall, research in this field has gradually shifted from idealized theoretical models to more practical applications. Early work mainly focused on relaxing network assumptions, extending the range of objective functions, and introducing acceleration techniques. The move toward practical deployment has brought greater attention to constraints that are often absent from idealized models. While existing methods based on stochastic optimization, communication compression, and asynchronous updates have improved the practicality of distributed algprithms, several challenges still remain. Future research may explore the following three directions:

(1) Coupling and trade-offs of multiple constraints. Existing work commonly treats different challenges in isolation, while practical distributed systems are often affected by multiple interacting factors, including data noise, unreliable communication, and limited resources. It would be useful to examine the interactions among different constraints and develop more efficient distributed algorithms that remain effective when resources are limited.

(2) Privacy and security in distributed computing. Frequent information exchange among nodes inherently increases the risk of exposing sensitive raw data. Although differential privacy techniques have been extensively utilized to provide privacy guarantees, their injected noise may limit the convergence accuracy of algorithms. Accordingly, designing highly secure and reliable distributed algorithms will be a core prerequisite for driving distributed optimization toward secure and trustworthy application scenarios.

(3) Distributed optimization for Large Language Models (LLMs). As artificial intelligence officially enters the era of LLMs, the volume of model parameters frequently scales into the billions or even trillions. In such high-dimensional settings, traditional methods face significant challenges arising from the curse of dimensionality. Future research can consider the deep integration of distributed optimization with low-rank subspace optimization techniques, such as parameter-efficient fine-tuning (PEFT). This will facilitate the design of novel algorithms that are memory-friendly and highly aligned with the network characteristics of large models.

Funding

This work was supported in part by the Natural Science Foundation of China under grants 62576053 and 62302068.

Conflicts of Interest

The author declares no conflict of interest.

Use of AI and AI-assisted Technologies

During the preparation of this manuscript, the authors used Gemini to refine the language and improve readability. After employing this tool, the authors carefully reviewed and edited the content as necessary and take full responsibility for the final published version of the survey.

References

1. Liu, J.; Ma, E.; Zha, L.; et al. Distributed Energy Management for Microgrids: When Privacy Preservation Meets Multiple Mechanisms. *IEEE Trans. Consum. Electron.* **2026**, *72*, 1514–1523.
2. Wang, J.; Hong, Y.; Wang, J.; et al. Cooperative and Competitive Multi-Agent Systems: From Optimization to Games. *IEEE/CAA J. Autom. Sin.* **2022**, *9*, 763–783.
3. Gao, C.; Zhang, Q.; Varshney, P.K.; et al. A Distributed Multi-Objective Detection Method for Multi-Sensor Systems with Unknown Local SNR. *IEEE Trans. Signal Process.* **2025**, *73*, 649–663.
4. Strinati, E.C.; Alexandropoulos, G.C.; Amani, N.; et al. Toward Distributed and Intelligent Integrated Sensing and Communications for 6G Networks. *IEEE Wireless Commun.* **2025**, *32*, 60–67.
5. Lü, Q.; Li, H.; Wu, C.; et al. Decentralized Constrained Optimization Over Time-Varying Directed Networks via Subgradient Rescaling. *IEEE Trans. Cybern.* **2026***56*, 3725–3738.
6. Nedic, A.; Ozdaglar, A. Distributed Subgradient Methods for Multi-Agent Optimization. *IEEE Trans. Autom. Control* **2009**, *54*, 48–61.

7. Shi, W.; Ling, Q.; Wu, G.; et al. Extra: An Exact First-Order Algorithm for Decentralized Consensus Optimization. *SIAM J. Optim.* **2015**, *25*, 944–966.
8. Ghaderyan, D.; Aybat, N.S.; Aguiar, A.P. A Fast Row-Stochastic Decentralized Method for Distributed Optimization Over Directed Graphs. *IEEE Trans. Autom. Control* **2024**, *69*, 275–289.
9. Zheng, Y.; Cenedese, C.; Cucuzzella, M.; et al. A Specified-Time Distributed Optimization Algorithm for Multi-Agent Systems With Time-Varying Constraints. *IEEE Trans. Autom. Control* **2025**, *71*, 1999–2006.
10. Ou, M.; Zhang, H.; Yan, Z.; et al. A Novel Distributed Gradient Algorithm for Composite Constrained Optimization Over Directed Network. *Sci. Rep.* **2026**, *16*, 5185.
11. Li, J.; Li, C.; Fan, J.; et al. Online Distributed Stochastic Gradient Algorithm for Nonconvex Optimization with Compressed Communication. *IEEE Trans. Autom. Control* **2023**, *69*, 936–951.
12. Rahili, S.; Ren, W. Distributed Continuous-Time Convex Optimization with Time-Varying Cost Functions. *IEEE Trans. Autom. Control* **2016**, *62*, 1590–1605.
13. Liu, T.; Bin, M.; Notarnicola, I.; et al. Robust Distributed Optimization for Multi-Agent Systems with General Linear Models. *IEEE Trans. Autom. Control* **2026**, *71*, 6685–6697.
14. Zhou, X.; Ma, Z.; Zou, S.; et al. Distributed Momentum-Based Multiagent Optimization with Different Constraint Sets. *IEEE Trans. Autom. Control* **2024**, *70*, 963–978.
15. Liao, F.Y.; Madden, T.; Zheng, Y. An Accelerated Proximal Bundle Method for Convex Optimization. *arXiv* **2025**, arXiv:2512.04523.
16. Gao, R.; Kleywegt, A. Distributionally Robust Stochastic Optimization with Wasserstein Distance. *Math. Oper. Res.* **2023**, *48*, 603–655.
17. Horvath, S.; Kovalev, D.; Mishchenko, K.; et al. Stochastic Distributed Learning with Gradient Quantization and Double-Variance Reduction. *Optim. Methods Softw.* **2023**, *38*, 91–106.
18. Wu, X.; Liu, C.; Magnússon, S.; et al. Asynchronous Distributed Optimization with Delay-Free Parameters. *IEEE Trans. Autom. Control* **2025**, *71*, 259–274.
19. Zheng, Y.; Liu, Q. A Review of Distributed Optimization: Problems, Models and Algorithms. *Neurocomputing* **2022**, *483*, 446–459.
20. Yang, T.; Yi, X.; Wu, J.; et al. A Survey of Distributed Optimization. *Annu. Rev. Control* **2019**, *47*, 278–305.
21. Li, X.; Xie, L.; Li, N. A Survey on Distributed Online Optimization and Online Games. *Annu. Rev. Control* **2023**, *56*, 100904.
22. AbdelRaouf, H.; Hasnaine, Q.R.; Fouda, M.M.; et al. Paradigm Shift toward Distributed Learning in IoT Intelligence: A Comprehensive Survey of Opportunities and Challenges. *IEEE Internet Things J.* **2026**, *13*, 20128–20154.
23. Vandenberghe, L.; Boyd, S. *Convex Optimization*, 1st ed.; Cambridge University Press: Cambridge, UK, 2004.
24. Sawamura, R.; Hayashi, N.; Inuiguchi, M. A Distributed Primal–Dual Push-Sum Algorithm on Open Multiagent Networks. *IEEE Trans. Autom. Control* **2024**, *70*, 1192–1199.
25. Xi, C.; Khan, U.A. DEXTRA: A Fast Algorithm for Optimization Over Directed Graphs. *IEEE Trans. Autom. Control* **2017**, *62*, 4980–4993.
26. Xi, C.; Xin, R.; Khan, U.A. ADD-OPT: Accelerated Distributed Directed Optimization. *IEEE Trans. Autom. Control* **2017**, *63*, 1329–1339.
27. Nedic, A.; Olshevsky, A.; Shi, W. Achieving Geometric Convergence for Distributed Optimization Over Time-Varying Graphs. *SIAM J. Optim.* **2017**, *27*, 2597–2633.
28. Xin, R.; Xi, C.; Khan, U.A. FROST—Fast Row-Stochastic Optimization with Uncoordinated Step-Sizes. *EURASIP J. Adv. Signal Process.* **2019**, *2019*, 1–14.
29. Pu, S.; Shi, W.; Xu, J.; et al. Push–Pull Gradient Methods for Distributed Optimization in Networks. *IEEE Trans. Autom. Control* **2020**, *66*, 1–16.
30. Xin, R.; Khan, U.A. A Linear Algorithm for Optimization Over Directed Graphs with Geometric Convergence. *IEEE Control Syst. Lett.* **2018**, *2*, 315–320.
31. Saadatniaki, F.; Xin, R.; Khan, U.A. Decentralized Optimization Over Time-Varying Directed Graphs with Row and Column-Stochastic Matrices. *IEEE Trans. Autom. Control* **2020**, *65*, 4769–4780.
32. Nedic, A.; Nguyen, D.T.A.; Nguyen, D.T. AB/Push-Pull Method for Distributed Optimization in Time-Varying Directed Networks. *Optim. Methods Softw.* **2025**, *40*, 1044–1071.
33. 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.
34. Shi, W.; Ling, Q.; Yuan, K.; et al. On the Linear Convergence of the ADMM in Decentralized Consensus Optimization. *IEEE Trans. Signal Process.* **2014**, *62*, 1750–1761.
35. Aybat, N.S.; Wang, Z.; Lin, T.; et al. Distributed Linearized Alternating Direction Method of Multipliers for Composite Convex Consensus Optimization. *IEEE Trans. Autom. Control* **2017**, *63*, 5–20.
36. Chang, T.H.; Hong, M.; Wang, X. Multi-Agent Distributed Optimization via Inexact Consensus ADMM. *IEEE Trans. Signal Process.* **2014**, *63*, 482–497.

37. Shi, W.; Ling, Q.; Wu, G.; et al. A Proximal Gradient Algorithm for Decentralized Composite Optimization. *IEEE Trans. Signal Process.* **2015**, *63*, 6013–6023.
38. Di Lorenzo, P.; Scutari, G. NEXT: In-Network Nonconvex Optimization. *IEEE Trans. Signal Inf. Process. Netw.* **2016**, *2*, 120–136.
39. Zeng, J.; Yin, W. On Nonconvex Decentralized Gradient Descent. *IEEE Trans. Signal Process.* **2018**, *66*, 2834–2848.
40. Zeng, J.; He, T.; Wang, M. A Fast Proximal Gradient Algorithm for Decentralized Composite Optimization Over Directed Networks. *Syst. Control Lett.* **2017**, *107*, 36–43.
41. Sun, Y.; Scutari, G.; Palomar, D. Distributed Nonconvex Multiagent Optimization Over Time-Varying Networks. In Proceedings of the 50th Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 6–9 November 2016.
42. Xin, R.; Das, S., Khan, U. A., et al. A Stochastic Proximal Gradient Framework for Decentralized Non-Convex Composite Optimization: Topology-Independent Sample Complexity and Communication Efficiency. *arXiv* **2021**, arXiv:2110.01594.
43. Xing, M.; Ma, D.; Ma, K.; et al. Implicit Enhanced Distributed Heavy-Ball Energy Management Strategy for Microgrids with Time-Varying Social Welfare and Delay. *IEEE Trans. Smart Grid* **2025**, *16*, 2492–2503.
44. Xue, S.; Tan, J.; Cao, H.; et al. Nesterov Accelerated Gradient-Based Fixed-Time Convergent Actor-Critic Control for Nonlinear Systems. *IEEE Trans. Autom. Sci. Eng.* **2026**, *23*, 5075–5083.
45. Qu, G.; Li, N. Accelerated Distributed Nesterov Gradient Descent. *IEEE Trans. Autom. Control* **2019**, *65*, 2566–2581.
46. Yang, Z.; He, W.; Aguiar, A.P. An Accelerated Distributed Proportional-Integral Method for Distributed Constrained Optimization. *IEEE Control Syst. Lett.* **2025**, *9*, 2241–2246.
47. 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.
48. Xin, R.; Jakovetić, D.; Khan, U.A. Distributed Nesterov Gradient Methods Over Arbitrary Graphs. *IEEE Signal Process. Lett.* **2019**, *26*, 1247–1251.
49. Li, H.; Cheng, H.; Wang, Z.; et al. Distributed Nesterov Gradient and Heavy-Ball Double Accelerated Asynchronous Optimization. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 5723–5737.
50. Luo, B.; Lü, Q.; Cheng, H.; et al. A Distributed Stochastic Unified Accelerated Algorithm for Convex Optimization Over Time-Varying Directed Networks. *IEEE Trans. Syst. Man Cybern. Syst.* **2026**, *56*, 4467–4480.
51. Pu, S.; Nedic, A. Distributed Stochastic Gradient Tracking Methods. *Math. Program.* **2021**, *187*, 409–457.
52. Gao, J.; Liu, X.W.; Dai, Y.H.; et al. Distributed Stochastic Gradient Tracking Methods with Momentum Acceleration for Non-Convex Optimization. *Comput. Optim. Appl.* **2023**, *84*, 531–572.
53. Nguyen, D.T.A.; Nguyen, D.T.; Nedic, A. Accelerated AB/Push–Pull Methods for Distributed Optimization Over Time-Varying Directed Networks. *IEEE Trans. Control Netw. Syst.* **2023**, *11*, 1395–1407.
54. Mohammadi, H.; Razaviyayn, M.; Jovanović, M.R. Tradeoffs Between Convergence Rate and Noise Amplification for Momentum-Based Accelerated Optimization Algorithms. *IEEE Trans. Autom. Control* **2024**, *70*, 889–904.
55. Nazari, P.; Tarzanagh, D.A.; Michailidis, G. Dadam: A Consensus-Based Distributed Adaptive Gradient Method for Online Optimization. *IEEE Trans. Signal Process.* **2022**, *70*, 6065–6079.
56. Carnevale, G.; Farina, F.; Notarnicola, I.; et al. GTAdam: Gradient Tracking with Adaptive Momentum for Distributed Online Optimization. *IEEE Trans. Control Netw. Syst.* **2022**, *10*, 1436–1448.
57. Wang, Z.; Ling, Q.; Yin, W. Decentralized Bundle Method for Nonsmooth Consensus Optimization. In Proceedings of the 2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP), Montreal, QC, Canada, 14–16 November 2017.
58. Zhu, Z.; Tian, Y.P.; Wu, X. Historical Information Accelerates Decentralized Optimization: A Proximal Bundle Method. *arXiv* **2025**, arXiv:2512.15189.
59. Liu, L.; Huang, Y.; Deng, C. Accelerated Distributed Cooperative Energy Management for Integrated Energy Systems. *IEEE/CAA J. Autom. Sin.* **2025**, *12*, 2589–2601.
60. Wang, X.; Wang, G.; Li, S. A Distributed Fixed-Time Optimization Algorithm for Multi-Agent Systems. *Automatica* **2020**, *122*, 109289.
61. Firouzbahrami, M.; Nobakhti, A. Cooperative Fixed-Time/Finite-Time Distributed Robust Optimization of Multi-Agent Systems. *Automatica* **2022**, *142*, 110358.
62. Chen, S.; Jiang, H.; Yu, Z. A Distributed Prescribed-Time Optimization Analysis for Multi-Agent Systems. *Inf. Sci.* **2022**, *607*, 346–360.
63. Li, H.; Zhang, M.; Yin, Z.; et al. Prescribed-Time Distributed Optimization Problem with Constraints. *ISA Transactions* **2024**, *148*, 255–263.
64. Pu, S.; Olshevsky, A.; Paschalidis, I.C. A Sharp Estimate on the Transient Time of Distributed Stochastic Gradient Descent. *IEEE Trans. Autom. Control* **2021**, *67*, 5900–5915.
65. Xin, R.; Khan, U.A.; Kar, S. An Improved Convergence Analysis for Decentralized Online Stochastic Non-Convex Optimization. *IEEE Trans. Signal Process.* **2021**, *69*, 1842–1858.
66. Chen, Y.; Hashemi, A.; Vikalo, H. Accelerated Distributed Stochastic Nonconvex Optimization Over Time-Varying Directed Networks. *IEEE Trans. Autom. Control* **2024**, *70*, 2196–2211.

67. Defazio, A.; Bach, F.; Lacoste-Julien, S. SAGA: A Fast Incremental Gradient Method with Support for Non-Strongly Convex Composite Objectives. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 1646–1654.
68. Johnson, R.; Zhang, T. Accelerating Stochastic Gradient Descent Using Predictive Variance Reduction. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 315–323.
69. Xin, R.; Khan, U.A.; Kar, S. Variance-Reduced Decentralized Stochastic Optimization with Accelerated Convergence. *IEEE Trans. Signal Process.* **2020**, *68*, 6255–6271.
70. Xin, R.; Khan, U.A.; Kar, S. A Fast Randomized Incremental Gradient Method for Decentralized Nonconvex Optimization. *IEEE Trans. Autom. Control* **2021**, *67*, 5150–5165.
71. Niu, Y.; Li, H.; Wang, Z.; et al. A Distributed Stochastic Proximal-Gradient Algorithm for Composite Optimization. *IEEE Trans. Control Netw. Syst.* **2021**, *8*, 1383–1393.
72. Craciunas, S.S.; Oliver, R.S. Combined Task-and Network-Level Scheduling for Distributed Time-Triggered Systems. *Real-Time Syst.* **2016**, *52*, 161–200.
73. Zhang, X.; Zhao, X. Architecture Design of Distributed Redundant Flight Control Computer Based on Time-Triggered Buses for UAVs. *IEEE Sensors J.* **2020**, *21*, 3944–3954.
74. Hayashi, N.; Sugiura, T.; Kajiyama, Y.; et al. Event-Triggered Consensus-Based Optimization Algorithm for Smooth and Strongly Convex Cost Functions. In Proceedings of the 2018 IEEE Conference on Decision and Control (CDC), Miami Beach, FL, USA, 17–19 December 2018.
75. Hu, C.; Zhang, X.; Wu, Q. Collaborative Active and Reactive Power Control of DERs for Voltage Regulation and Frequency Support by Distributed Event-Triggered Heavy Ball Method. *IEEE Trans. Smart Grid* **2023**, *14*, 3804–3815.
76. Xu, L.; Yi, X.; Shi, Y.; et al. Distributed Nonconvex Optimization with Event-Triggered Communication. *IEEE Trans. Autom. Control* **2023**, *69*, 2745–2752.
77. Liu, L.; Yang, G.H. Distributed Dynamic Event-Triggered Optimal Economic Dispatch for Multienergy Systems with Prescribed-Time Convergence. *IEEE Trans. Ind. Inform.* **2025**, *21*, 4533–4542.
78. Zhang, J.; You, K.; Başar, T. Distributed Discrete-Time Optimization in Multiagent Networks Using Only Sign of Relative State. *IEEE Trans. Autom. Control* **2018**, *64*, 2352–2367.
79. Aji, A.F.; Heafield, K. Sparse Communication for Distributed Gradient Descent. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP), Copenhagen, Denmark, 7–11 September 2017.
80. Alistarh, D.; Grubic, D.; Li, J.; et al. QSGD: Communication-Efficient SGD via Gradient Quantization and Encoding. In Proceedings of the 31st Conference on Neural Information Processing Systems (NeurIPS), Long Beach, CA, USA, 4–9 December 2017.
81. Koloskova, A.; Stich, S.; Jaggi, M. Decentralized Stochastic Optimization and Gossip Algorithms with Compressed Communication. In Proceedings of the 36th International Conference on Machine Learning (ICML), Long Beach, CA, USA, 9–15 June 2019.
82. Lü, Q.; Guan, M.; He, C.; et al. A Quantised Push-Sum Distributed Adaptive Momentum Algorithm for Optimisation Over Directed Networks. *CAAI Trans. Intell. Technol.* **2026**, *11*, 1284–1301.
83. Yi, X.; Zhang, S.; Yang, T.; et al. Communication Compression for Distributed Nonconvex Optimization. *IEEE Trans. Autom. Control* **2022**, *68*, 5477–5492.
84. Liu, W.; Panda, A.; Pandey, U.; et al. Compressed Decentralized Momentum Stochastic Gradient Methods for Nonconvex Optimization. *arXiv* **2025**, arXiv:2508.04950.
85. Li, H.; Lü, Q.; Chen, G.; et al. Distributed Constrained Optimization Over Unbalanced Directed Networks Using Asynchronous Broadcast-Based Algorithm. *IEEE Trans. Autom. Control* **2020**, *66*, 1102–1115.
86. Iutzeler, F.; Bianchi, P.; Ciblat, P.; et al. Asynchronous Distributed Optimization Using a Randomized Alternating Direction Method of Multipliers. In Proceedings of the 52nd IEEE Conference on Decision and Control (CDC), Firenze, Italy, 10–13 December 2013.
87. Wu, X.; Liu, C.; Magnusson, S.; et al. Delay-Agnostic Asynchronous Distributed Optimization. In Proceedings of the 62nd IEEE Conference on Decision and Control (CDC), Singapore, 13–15 December 2023.
88. Tian, Y.; Sun, Y.; Scutari, G. Achieving Linear Convergence in Distributed Asynchronous Multiagent Optimization. *IEEE Trans. Autom. Control* **2020**, *65*, 5264–5279.
89. Nie, R.; Du, W.; Wang, T.; et al. Distributed Asynchronous Optimization of Multiagent Systems: Convergence Analysis and Its Application. *IEEE Trans. Ind. Inform.* **2024**, *20*, 8983–8992.
90. Su, E.; Li, H.; Zheng, Z.; et al. Distributed Constrained Optimization with Asynchrony and Delays. *IEEE Trans. Circuits Syst. II Express Briefs* **2021**, *69*, 2772–2776.
91. Ran, L.; Li, H.; Zheng, L.; et al. Asynchronous Distributed Nonsmooth Composite Optimization via Computation-Efficient Primal-Dual Proximal Algorithms. *IEEE Trans. Emerg. Top. Comput. Intell.* **2024**, *9*, 1595–1609.
92. Kungurtsev, V.; Morafah, M.; Javidi, T.; et al. Decentralized Asynchronous Nonconvex Stochastic Optimization on Directed Graphs. *IEEE Trans. Control Netw. Syst.* **2023**, *10*, 1796–1804.