Distributed Optimization in Multi-agent Networks Using One-bit of Relative State Information

Jiaqi Zhang and Keyou You

Abstract This chapter is concerned with the design of distributed discrete-time algorithms to cooperatively solve an additive cost optimization problem in multi-agent networks. The striking feature of our distributed algorithms lies in the use of only the sign of relative state information between neighbors, which substantially differentiates our algorithms from others in the existing literature. Moreover, the algorithm does not require the interaction matrix to be doubly-stochastic. We first interpret the proposed algorithms in terms of the penalty method in optimization theory and then perform non-asymptotic analysis to study convergence for static network graphs. Compared with the celebrated distributed subgradient algorithms, which however use the exact relative state information, the convergence speed is essentially not affected by the loss of information. We also extend our results to the cases of deterministically and randomly time-varying graphs. Finally, we validate the theoretical results by simulations.

1 Introduction

In recent years, distributed optimization problems in multi-agent systems have attracted increasing attention. Distributed optimization is concerned with that all agents to cooperatively minimize a sum of local objective functions over a graph. The key feature of such an optimization problem lies in that each agent only knows a local component of the objective function and thus must cooperate with its neighbors to compute the optimal value. The interaction between nodes is modeled by an algebraic graph. The motivating examples for distributed computation include the AUV

Parts of the results in this chapter have previously been appeared in [26–28].

J. Zhang · K. You (✉)
Department of Automation, and BNRist, Tsinghua University, Beijing 100084, China
e-mail: youky@tsinghua.edu.cn

J. Zhang
e-mail: zjq16@mails.tsinghua.edu.cn

T. Başar (ed.), *Uncertainty in Complex Networked Systems*,
Systems & Control: Foundations & Applications,
https://doi.org/10.1007/978-3-030-04630-9_13

formation control [24], large-scale machine learning [4, 17, 25], and the distributed quantile regression over sensor networks [21].

To solve the distributed optimization problem, the majority of the algorithms (see e.g., [11, 15, 16, 21] and the references therein) are generally comprised of two parts. One is to drive all agents to consensus, which is based on the well-known consensus algorithm [18]. The other one is to push the consensus value toward an optimal point by using the local (sub)gradient in each node. In this case, subgradient-based algorithms have been widely used. To achieve consensus of the multi-agent network, most of the existing methods require each agent to access the state values of its neighbors at each time, either exactly [15, 18] or in a quantized form [19, 23]. However, in some situations, an agent may only roughly know relative state measurements between its neighbors. For example, consider the case of several robots working in a plane, when each robot can only tell which quadrant its neighbor is lying by cheap sensors but not the neighbor's accurate relative position. Thus, the information accessible is restricted to be only one bit. Note that this is different from the quantized setting in [19], which studied the effects of exchanging a quantized rather than an exact state between neighbors. This is also different from previous studies on exchanging quantized gradients [13] since we are only using the quantized relative state information. Therefore, most algorithms in the literature, particularly the ones in the references cited above, cannot handle the case of one-bit information. It is worth noting that another advantage of our algorithm, in addition to using only one bit of relative information, is that it does not require the interaction matrix of the agents to be doubly-stochastic. A doubly-stochastic adjacency matrix is a common assumption in many existing algorithms [14, 16, 20], but it is restrictive in the distributed setting. For example, the Metropolis method [20] to construct a doubly-stochastic matrix requires each node to know its neighbors' degrees, which may be impractical in applications, especially when the graph is time-varying.

Designing an algorithm using one bit of information often involves nonlinear systems analysis, which is substantially different from the commonly applied graph Laplacian theory in the aforementioned works. There are, however, some exceptions [5, 9, 12]. In [5], the authors designed a consensus algorithm using only sign information of the relative state. A similar algorithm was also proposed in [9] to distributedly compute a sample median. The algorithm in [12] is the most relevant to the one in this chapter except that it is a continuous-time algorithm, which adopts a completely different analysis tool than ours. We will return to this point, and discuss more extensively later.

In fact, all the aforementioned works that use one bit of information focused on continuous-time algorithms. However, a discrete-time algorithm is worth studying because many distributed optimization applications involve communication between agents and control of agents, which are typically discrete in nature. Besides, a discrete-time algorithm is easier to implement. What is more, a continuous-time algorithm cannot be extended to the discrete-time case that easily, since the methods used to analyze continuous-time algorithms in the above works are often based on Lyapunov theory. We know that some general stepsize rules (e.g., constant, diminishing) in discrete-time gradient-based algorithms cannot guarantee the nonincreas-

ingness of a latent Lyapunov function, and some special stepsize rules (e.g., line minimization rule) often fail to meet the requirement of distributed computation, which renders the Lyapunov analysis difficult to extend to the discrete-time case. Therefore, an alternative method is urgently needed, which is what this chapter does.

More precisely, we propose in this chapter a distributed optimization algorithm using only one bit of information in the discrete-time case. Different from most of the previous works, our analysis is based on optimization theory rather than algebraic graph theory or Lyapunov theory. There are two underlying advantages of this. First, compared to many existing approaches which first propose an algorithm, and then find a Lyapunov function to prove its convergence, the intuition behind our algorithm appears to be more natural and reasonable, as it aims to minimize a well-designed objective function. Second, a wealth of research in convex optimization theory ensures our algorithm more easily extensible to more general cases. For example, our algorithm over time-varying graphs is a direct extension of that over static graphs. Specifically, we extend our algorithm to both *deterministically* time-varying graphs and *randomly* time-varying graphs. The former can model the time-varying topology of agents in applications [17, 22], while the latter can be used to describe the gossip networks [10], random package losses in communication networks, etc. Based on optimization theory, our methods to analyze the cases of deterministically time-varying graphs and randomly time-varying graphs take advantage of incremental gradient methods and stochastic gradient descent methods, respectively.

For a connected static graph, each node of the distributed optimization algorithm is shown to converge asymptotically to the same optimal point of the optimization without any reduction in the convergence rate. For deterministically time-varying graphs, the convergence of the distributed optimization algorithm is established if the graphs are uniformly jointly connected. For randomly time-varying graphs, we show the convergence of the distributed optimization algorithm in the almost sure sense under the so-called randomly activated connected graph assumption.

The rest of the chapter is organized as follows. Section 2 provides some preliminaries and introduces the distributed optimization problem. In Sect. 3, we present our discrete-time distributed optimization algorithm using one bit of information. Section 4 includes our main results on convergence and convergence rate of the algorithm over static graphs. Section 5 provides the convergence results over uniformly jointly connected graphs and randomly activated graphs. Finally, we perform simulations to validate the theoretical results in Sect. 6, and draw some concluding remarks in Sect. 7.

Notation: We use a, $\mathbf{a}$, A, and $\mathscr{A}$ to denote a scalar, vector, matrix, and set, respectively. $\mathbf{a}^\mathsf{T}$ and A^T denote the transposes of $\mathbf{a}$ and A, respectively. $[A]_{ij}$ denotes the element in row i and column j of A. $\mathbb{R}$ denotes the set of real numbers and $\mathbb{R}^n$ denotes the set of all n-dimensional real vectors. $\mathbb{1}$ denotes the vector with all ones, the dimension of which depends on the context. We let $\|\cdot\|_1$, $\|\cdot\|$ and $\|\cdot\|_\infty$ denote the l_1-norm, l_2-norm and l_∞-norm of a vector or matrix, respectively. We define

$$\operatorname{sgn}(x) = \begin{cases} 1, & \text{if } x \geq 0, \\ -1, & \text{otherwise.} \end{cases}$$

With a slight abuse of notation, $\nabla f(x)$ denotes *any* subgradient of $f(x)$ at x, i.e., $\nabla f(x)$ satisfies

$$f(y) \geq f(x) + (y - x)^{\mathsf{T}} \nabla f(x), \ \forall y \in \mathbb{R}. \tag{1}$$

The subdifferential $\partial f(x)$ is the set of all subgradients of $f(x)$ at x. If $f(x)$ is differentiable at x, then $\partial f(x)$ includes only the gradient of $f(x)$ at x.

We call $\inf_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x})$ the optimal value of $f(\mathbf{x})$. Any element from the set $\arg\inf_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x})$ is called an optimal solution or optimal point of $f(\mathbf{x})$.

Superscripts are usually used to represent sequence indices, i.e., x^k represents the value of the sequence $\{x^k\}$ at time k.

2 Problem Formulation

This section introduces some basics of graph theory, and presents the distributed optimization problem in multi-agent networks.

2.1 Basics of Graph Theory

A graph (network) is represented as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, ..., n\}$ is the set of nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges. Let $\mathcal{N}_i = \{j \in \mathcal{V} | (i, j) \in \mathcal{E}\}$ be the set of neighbors of node i, and $A = [a_{ij}]$ be the weighted adjacency matrix of $\mathcal{G}$, where $a_{ij} > 0$ if and only if there exists an edge connecting nodes i and j, and otherwise, $a_{ij} = 0$. If $A = A^{\mathsf{T}}$, the associated graph is undirected. This chapter focuses only on undirected graphs.

In the case of time-varying graphs, we use $\mathcal{G}^k = (\mathcal{V}, \mathcal{E}^k, A^k)$ to represent the graph at time k. Let $\mathcal{G}^{k_1} \cup \mathcal{G}^{k_2}$ denote the graph $(\mathcal{V}, \mathcal{E}^{k_1} \cup \mathcal{E}^{k_2}, A^{k_1} + A^{k_2})$. Let $\mathcal{N}_i^k = \{j \in \mathcal{V} | (i, j) \in \mathcal{E}^k\}$ denote the set of neighbors of node i at time k. The incidence matrix $B \in \mathbb{R}^{n\times m}$ of $\mathcal{G}$ is defined by

$$B_{ie} = \begin{cases} 1, \text{ if node } i \text{ is the source node of edge } e, \\ -1, \text{ if node } i \text{ is the sink node of edge } e, \\ 0, \text{ otherwise.} \end{cases}$$

For any $\mathbf{x} = [x_1, ..., x_n]^{\mathsf{T}}$, we have that

$$\mathbf{b}_e^{\mathsf{T}} \mathbf{x} = x_i - x_j$$

where $\mathbf{b}_e$, $e \in \mathcal{E}$ is the e-th column of B, and i and j are the source and the sink nodes of edge e, respectively.

A path is a sequence of consecutive edges that connect a set of different nodes. We say a graph is *connected* if there exists a path between any pair of two nodes. To evaluate the intensity of the graph's connectivity, we introduce an important concept called l-connected graph below.

Definition 1 (l-*connected graph*) A connected graph is l-connected ($l \geq 1$) if it remains connected whenever fewer than l edges are removed.

Clearly, a connected graph is at least 1-connected and each node of an l-connected graph has at least l neighbors.

2.2 Distributed Optimization Problem

With only the sign of relative state, our objective is to distributedly solve the multi-agent optimization problem

$$\underset{x\in\mathbb{R}}{\text{minimize}} \ f(x) := \sum_{i=1}^{n} f_i(x) \tag{2}$$

where for each $i \in \mathcal{V}$, the local objective function $f_i(x)$ is continuously convex but not necessarily differentiable, and is only known by node i. The number of nodes is set to be $n > 1$. We first make a mild assumption.

Assumption 1 (Nonempty optimal set and bounded subgradients)

(a) The set $\mathcal{X}^\star$ of optimal solutions of problem (2) is nonempty, i.e., for any $x^\star \in \mathcal{X}^\star$, it holds that $f^\star := f(x^\star) = \inf_{x\in\mathbb{R}} f(x)$.
(b) There exists a constant $c > 0$ such that

$$|\nabla f_i(x)| \leq c, \ \forall i \in \mathcal{V}, x \in \mathbb{R}. \tag{3}$$

Assumption 1 is common in the literature, see e.g., [16, 25]. In particular, the second part is often made to guarantee the convergence of a subgradient method [16], and obviously holds if the decision variable x is restricted to a compact set.

3 The Distributed Optimization Algorithm Over Static Graphs

In this section, we provide the discrete-time distributed optimization algorithm that uses only the sign information of the relative state of the neighboring nodes (hence one-bit information), and then interpret it via the penalty method in optimization theory.

This section only focuses on static graphs, which are important to the analysis of time-varying cases in following sections.

3.1 The Distributed Optimization Algorithm

The discrete-time distributed algorithm to solve (2) over a static network $\mathscr{G}$ is given in Algorithm 1.

Algorithm 1: Distributed Algorithm Using the Sign of Relative State

1: **Initialization**: Every node i sets $x_i^0 = 0$ for all $i \in \mathscr{V}$.
2: **Repeat**
3: **Information collection**: Each node i collects the sign of the relative state to its neighbor $j \in \mathscr{N}_i$ and obtain r_i^k, which is given below

$$r_i^k = \sum_{j \in \mathscr{N}_i} a_{ij} \operatorname{sgn}(x_j^k - x_i^k).$$

4: **Local update**: The decision variable in each node is locally updated as

$$x_i^{k+1} = x_i^k + \rho^k \left(\lambda \cdot r_i^k - \nabla f_i(x_i^k) \right),$$

where λ and ρ^k are to be given, and $\nabla f_i(x_i^k)$ is any subgradient of $f_i(x)$ at x_i^k.
5: **Set** $k = k + 1$.
6: **Until** a predefined stopping rule (e.g., a maximum iteration number) is satisfied.

The continuous-time version of Algorithm 1 is also given in (4) of [12] and is proved to be convergent by using the non-smooth analysis tool [6]. To ensure a valid algorithm, it is important to choose both λ and ρ_k, which, for the discrete-time case, requires a completely different approach from that of [12], as it will be evident in Sect. 3.2.

Compared with the celebrated distributed gradient descent (DGD) algorithm, see e.g.,[16],

$$x_i^{k+1} = x_i^k + \sum_{j \in \mathscr{N}_i} \tilde{a}_{ij}(x_j^k - x_i^k) - \rho^k \nabla f_i(x_i^k). \tag{4}$$

Algorithm 1 has at least two advantages. First, each node i in Algorithm 1 only uses the binary information of the relative state $(x_j^k - x_i^k)$, instead of the exact relative state from each of its neighbors j, which is essential in some cases where $\operatorname{sgn}(x_j^k - x_i^k)$ is the only available information. Second, Algorithm 1 does not require the adjacency matrix A^k to be doubly-stochastic, while associated adjacency matrix $\tilde{A}^k$ must be

doubly-stochastic in DGD [16], where $[\tilde{A}^k]_{ij} := \tilde{a}_{ij}^k$. This is very restrictive in the distributed setting.

Remark 1 Algorithm 1 also works if x is a vector by applying $\mathrm{sgn}(\cdot)$ to each element of the relative state vector. All the results on the scalar case continue to hold with such an adjustment.

3.2 Penalty Method Interpretation of Algorithm 1

In this subsection, we interpret Algorithm 1 via the penalty method and show that it is the subgradient iteration of a penalized optimization problem.

Notice that problem (2) can be essentially reformulated as follows:

$$\begin{aligned} \underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad & g(\mathbf{x}) := \sum_{i=1}^{n} f_i(x_i) \\ \text{subject to} \quad & x_i = x_j,\ \forall i, j \in \{1, ..., n\} \end{aligned} \tag{5}$$

where $\mathbf{x} = [x_1, ..., x_n]^\mathsf{T}$. It is easy to see that the optimal value of problem (5) is also $f^\star$, and the set of optimal solutions is $\{x^\star \mathbb{1} | x^\star \in \mathscr{X}^\star\}$.

Define a penalty function by

$$h(\mathbf{x}) = \frac{1}{2} \sum_{i=1}^{n} \sum_{j\in\mathscr{N}_i} a_{ij} |x_i - x_j|. \tag{6}$$

If the associated network $\mathscr{G}$ is connected, then $h(\mathbf{x}) = 0$ is equivalent to that $x_i = x_j,\ \forall i, j \in \{1, ..., n\}$. Thus, a penalized optimization problem of (5) can be given as

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \ \ \widetilde{f}_\lambda(\mathbf{x}) := g(\mathbf{x}) + \lambda h(\mathbf{x}) \tag{7}$$

where $\lambda > 0$ is the penalty factor.

We show below that Algorithm 1 is just the subgradient iteration of the penalized problem (7) with stepsizes ρ^k. Recall that $\mathrm{sgn}(x)$ is a subgradient of $|x|$ for any $x \in \mathbb{R}$. It follows from (6) that a subgradient $\nabla h(\mathbf{x}) = [\nabla h(\mathbf{x})_1, ..., \nabla h(\mathbf{x})_n]^\mathsf{T}$ of $h(\mathbf{x})$ is given element-wise by

$$\nabla h(\mathbf{x})_i = \sum_{j\in\mathscr{N}_i} a_{ij} \mathrm{sgn}(x_i - x_j),\ i \in \mathscr{V}.$$

Similarly, a subgradient $\nabla g(\mathbf{x}) = [\nabla g(\mathbf{x})_1, ..., \nabla g(\mathbf{x})_n]^\mathsf{T}$ of $g(\mathbf{x})$ is given element-wise by $\nabla g(\mathbf{x})_i = \nabla f_i(x_i)$. Then, the i-th element of a subgradient of $\widetilde{f}_\lambda(\mathbf{x})$ is given

as

$$\nabla \widetilde{f}_\lambda(\mathbf{x})_i = \lambda \sum_{j\in\mathcal{N}_i} a_{ij}\mathrm{sgn}(x_i - x_j) + \nabla f_i(x_i),\, i \in \mathcal{V}.$$

Finally, the subgradient method for solving (7) is given as

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \rho^k \nabla \widetilde{f}_\lambda(\mathbf{x}^k),$$

which is exactly the vector form of the local update in Algorithm 1. By [2], it follows that the subgradient method converges to an optimal solution of problem (7) if ρ^k is appropriately chosen.

For a finite $\lambda > 0$, the optimization problems (5) and (7) are generally not equivalent. Under mild conditions, our main result shows that they actually become equivalent if the penalty factor λ is strictly greater than an explicit lower bound. To this end, we define

$$\begin{aligned} \bar{x} &= \frac{1}{n}\mathbb{1}^\mathsf{T}\mathbf{x}, \\ v(\mathbf{x}) &= \max_i(x_i) - \min_i(x_i), \end{aligned} \tag{8}$$

and let $a_{\min}^{(l)}$ be the sum of the l smallest edges' weights, i.e.,

$$a_{\min}^{(l)} = \sum_{e=1}^{l} a_{(e)} \tag{9}$$

where $a_{(1)}, a_{(2)}, \ldots$ are given as an ascending order of the positive weights a_{ij} for any edge $(i, j) \in \mathcal{E}$.

Theorem 1 (Lower bound for the penalty factor, [28]) *Suppose that Assumption 1 holds, and that the multi-agent network is l-connected. If the penalty factor is selected as*

$$\lambda > \underline{\lambda} := \frac{nc}{2a_{min}^{(l)}}, \tag{10}$$

where c and $a_{min}^{(l)}$ are defined in (3) and (9), then

(a) The optimization problems (2) and (7) are equivalent in the sense that the set of optimal solutions and optimal value of (7) are given by $\widetilde{\mathcal{X}}^\star = \{x^\star\mathbb{1} | x^\star \in \mathcal{X}^\star\}$ and $f^\star$, respectively.

(b) For any $\mathbf{x} \notin \{\alpha\mathbb{1} | \alpha \in \mathbb{R}\}$, it holds that

$$\|\nabla \widetilde{f}_\lambda(\mathbf{x})\|_\infty \geq \frac{2\lambda a_{min}^{(l)}}{n} - c.$$

Proof (of part (a)) Consider the inequalities below

$$\begin{aligned}\widetilde{f}_\lambda(\mathbf{x}) &= \lambda h(\mathbf{x}) + g(\mathbf{x} - \bar{x}\mathbb{1} + \bar{x}\mathbb{1}) \\ &\geq \lambda h(\mathbf{x}) + g(\bar{x}\mathbb{1}) + (\mathbf{x} - \bar{x}\mathbb{1})^{\mathsf{T}} \nabla g(\bar{x}\mathbb{1}) \\ &\geq \lambda h(\mathbf{x}) + f(\bar{x}) - \|\mathbf{x} - \bar{x}\mathbb{1}\| \|\nabla g(\bar{x}\mathbb{1})\|\end{aligned} \tag{11}$$

where the equality follows from the definition of $\widetilde{f}_\lambda(\mathbf{x})$, the first inequality is from (1), and the second inequality results from the Cauchy–Schwarz inequality [2] as well as the fact that $g(a\mathbb{1}) = f(a)$.

Then, we can show that

$$h(\mathbf{x}) \geq a_{\min}^{(l)} v(\mathbf{x}). \tag{12}$$

Since the multi-agent network is l-connected, it follows from Menger's theorem [8] that there exist at least l disjoint paths (two paths are disjoint if they have no common edge) between any two nodes of the graph. Therefore, letting $x_{\max}$ and $x_{\min}$ be two nodes associated with the maximum element and the minimum element of $\mathbf{x}$, respectively, we can find l disjoint paths from $x_{\max}$ to $x_{\min}$.

Let $x_{(p,1)}, \ldots, x_{(p,n_p)}$ denote the nodes of path p in order, where n_p is the number of nodes in path p, and $x_{(p,1)} = x_{\max}, x_{(p,n_p)} = x_{\min}$ for all $p \in \{1, \ldots, l\}$. Since these l paths are disjoint, it follows that

$$\begin{aligned}h(\mathbf{x}) &\geq \sum_{p=1}^{l} \sum_{i=1}^{n_p - 1} a_{(p,i,i+1)} |x_{(p,i)} - x_{(p,i+1)}| \\ &\geq \sum_{p=1}^{l} \sum_{i=1}^{n_p - 1} \min_i a_{(p,i,i+1)} |x_{(p,i)} - x_{(p,i+1)}| \\ &\geq \sum_{p=1}^{l} \min_i a_{(p,i,i+1)} \sum_{i=1}^{n_p - 1} (x_{(p,i)} - x_{(p,i+1)}) \\ &\geq \sum_{p=1}^{l} \min_i a_{(p,i,i+1)} (x_{\max} - x_{\min}) \geq a_{\min}^{(l)} v(\mathbf{x})\end{aligned} \tag{13}$$

where $a_{(p,i,i+1)}$ is the weight of the edge connecting nodes $x_{(p,i)}$ and $x_{(p,i+1)}$.

Letting $\tilde{x} = \frac{1}{2}(\max_i(x_i) + \min_i(x_i))$, we have

$$\begin{aligned}\|\mathbf{x} - \bar{x}\mathbb{1}\| \|\nabla g(\bar{x}\mathbb{1})\| &\leq \|\mathbf{x} - \tilde{x}\mathbb{1}\| \|\nabla g(\bar{x}\mathbb{1})\| \\ &\leq \sqrt{n} \|\mathbf{x} - \tilde{x}\mathbb{1}\|_\infty \cdot \sqrt{n} \|\nabla g(\bar{x}\mathbb{1})\|_\infty \\ &\leq \frac{nc}{2} v(\mathbf{x}).\end{aligned} \tag{14}$$

where the first inequality follows from the fact that $\bar{x}$ minimizes $\|\mathbf{x} - \alpha\mathbb{1}\|$ with respect to (w.r.t.) α for all $\mathbf{x}$. Equations (11), (12) and (14) jointly imply the following inequality

$$\widetilde{f}_\lambda(\mathbf{x}) - f^\star \geq f(\bar{x}) - f^\star + (\lambda a_{\min}^{(l)} - \frac{cn}{2})v(\mathbf{x}). \tag{15}$$

Since $\lambda > nc/(2a_{\min}^{(l)})$, $v(\mathbf{x}) \geq 0$, $\forall \mathbf{x} \in \mathbb{R}^n$ and $f(\bar{x}) \geq f^\star, \forall \bar{x} \in \mathbb{R}$, then the right-hand side of (15) is nonnegative. That is, $\widetilde{f}_\lambda(\mathbf{x}) \geq f^\star$ for all $\mathbf{x} \in \mathbb{R}^n$.

Moreover, it follows from (7) that $\widetilde{f}_\lambda(x^\star\mathbb{1}) = f^\star$ for any $x^\star \in \mathscr{X}^\star$, i.e., $\widetilde{f}_\lambda(\mathbf{x}) = f^\star$ for any $\mathbf{x} \in \widetilde{\mathscr{X}}^\star$. What remains to be shown is that $\widetilde{f}_\lambda(\mathbf{x}) > f^\star$ for all $\mathbf{x} \notin \widetilde{\mathscr{X}}^\star$, which includes

Case (a): $\mathbf{x} \neq \alpha\mathbb{1}$ for any $\alpha \in \mathbb{R}$,
Case (b): $\mathbf{x} = \alpha\mathbb{1}$ for some $\alpha \notin \mathscr{X}^\star$.

For Case (a), $v(\mathbf{x})$ is strictly positive, and hence we know that $\widetilde{f}_\lambda(\mathbf{x}) > f^\star$ from (15). For Case (b), we have $v(\mathbf{x}) = 0$. By (15) we have that $\widetilde{f}_\lambda(\mathbf{x}) \geq f(\bar{x}) = f(\alpha) > f^\star$. Thus, $\widetilde{f}_\lambda(\mathbf{x}) > f^\star$ for all $\mathbf{x} \notin \widetilde{\mathscr{X}}^\star$, which completes the proof of part (a).

The proof of part (b) is very involved and the interested readers are referred to [28] for details. □

Algorithm 1(b) can also be modified to deal with objective functions with unbounded subgradients, e.g., quadratic functions, see [28] for details. Theorem 1 provides a sufficient condition for the equivalence between problems (5) and (7), and allows us to focus only on problem (7). Notice that this result is nontrivial even though the penalty method has been widely studied in optimization theory [2]. For example, a well-known result is that the gap between the optimal values of the penalized problem (7) and the problem (5) gets smaller as λ becomes larger, which however cannot always guarantee the existence of a finite penalty factor λ to eliminate the gap. A large λ may have negative effects on the transient performance of Algorithm 1.

Remark 2 It is worth mentioning that (10) in Theorem 1 also holds for the multidimensional case if Assumption 1(b) is replaced with $\|\nabla f_i(\mathbf{x})\| \leq c$ for all i and $\mathbf{x}$.

In view of the duality theory [2], a potential lower bound for λ could be the absolute value of the associated Lagrange multiplier. However, a Lagrange multiplier usually cannot be obtained before solving its dual problem. Theorem 1 gives an explicit lower bound for λ in terms of the network size and its connectivity, and is tighter than the bounds in [9] and [12].

In fact, the lower bound can be tight in some cases as shown in the following example. Note that [9] does not consider a generic optimization problem.

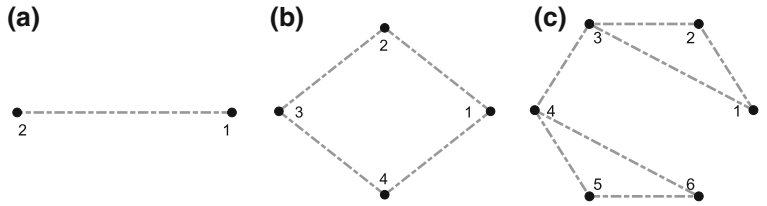

Fig. 1 Some graphs

Example 1 ([28]) Consider the graph in Fig. 1b with unit edge weights, i.e., $a_{ij} = 1$ for all $(i, j) \in \mathscr{V}$. Let $f_1(x) = |x|$, $f_2(x) = |x - 2|$, $f_3(x) = |x - 4|$, $f_4(x) = |x - 6|$ and $f(x) = \sum_{i=1}^{4} f_i(x)$. It is not difficult to compute that the optimal value of $f(x)$ is 8 and the set of optimal solutions is a closed interval $[2, 4]$. By (7), the corresponding penalized problem is given as

$$\begin{aligned}\widetilde{f}_\lambda(\mathbf{x}) = |x_1| + |x_2 - 2| + |x_3 - 4| + |x_4 - 6| + \\ \lambda(|x_1 - x_2| + |x_2 - x_3| + |x_3 - x_4| + |x_4 - x_1|).\end{aligned}$$

Theorem 1 implies that $\widetilde{f}_\lambda(\mathbf{x})$ has the same optimal value as $f(x)$ and the set of optimal solutions is $\widetilde{\mathscr{X}}^\star = \{x^\star \mathbb{1} | x^\star \in [2, 4]\}$, provided that $\lambda > 4 \cdot 1/(2 \cdot 2) = 1$.

Given any $\lambda \leq 1$, consider $\mathbf{x} = [2, 2, 4, 4]^\mathsf{T} \notin \widetilde{\mathscr{X}}^\star$. Clearly,

$$\widetilde{f}_\lambda(\mathbf{x}) = 4 + 4\lambda \leq f^\star = 8,$$

which implies that the set of optimal solutions of the penalized problem is not $\widetilde{\mathscr{X}}^\star$. Thus for any $\lambda \leq 1$, the original problem $f(x)$ cannot be solved via the penalized problem $\widetilde{f}_\lambda(\mathbf{x})$, and the lower bound in (10) is tight in this example. □

The lower bound in (10) is in a simple form and $a_{\min}^{(l)}$ cannot be easily replaced. One may consider to use the minimum degree of the network, i.e., $d_m = \min_{i \in \mathscr{V}} \sum_{j=1}^{n} a_{ij}$. This is impossible in some cases. Consider the 1-connected graph in Fig. 1c with unit edge weights. Then, $a_{\min}^{(1)} = 1$ and $d_m = 2$. Let $[s_1, ..., s_6] = [1, 2, 3, 4, 5, 6]$ and $f_i(x) = |x - s_i|, \ \forall i \in \{1, ..., 6\}$. Set

$$\mathbf{x} = [x_1, ..., x_6]^\mathsf{T} = [3, 3, 3, 4, 4, 4]^\mathsf{T}.$$

Then, using similar arguments as in Example 1, one can infer that the lower bound $\underline{\lambda}$ in (10) cannot be reduced to $nc/(2d_m) = 3/2$.

A similar penalty method interpretation of (4) with constant ρ^k is provided in [14], where the penalty function is chosen as

$$\mathbf{x}^\mathsf{T} L \mathbf{x} = \frac{1}{2} \sum_{i,j} a_{ij}(x_i - x_j)^2$$

and L is the graph Laplacian matrix. However, such a quadratic penalty function cannot always guarantee the existence of a finite λ for the equivalence of the two problems. We provide a concrete example to illustrate this.

Example 2 Consider the graph in Fig. 1a with unit edge weights. Let $f_1(x) = (x-1)^2$ and $f_2(x) = (x-3)^2$. Clearly, the optimal solution of $f(x) = f_1(x) + f_2(x)$ is $x^\star = 2$. Then a corresponding penalized problem using $x^\mathsf{T} L x$ is

$$\underset{x_1, x_2 \in \mathbb{R}}{\text{minimize}} \ f_1(x_1) + f_2(x_2) + \lambda (x_1 - x_2)^2. \tag{16}$$

The optimal solution of (16) is $x_1^\star = (1+4\lambda)/(1+2\lambda)$ and $x_2^\star = (3+4\lambda)/(1+2\lambda)$, and there does not exist a finite value of λ which makes both of them equal to $x^\star = 2$. □

By [2], $\mathbf{x}^\star$ is an optimal solution of (7) if and only if $0 \in \partial \widetilde{f}_\lambda(\mathbf{x}^\star)$. Part (b) of Theorem 1 shows that for any $\mathbf{x} \notin \{\alpha \mathbb{1} | \alpha \in \mathbb{R}\}$, the norm of the corresponding subgradient is uniformly greater than a positive lower bound, which clearly shows non-optimality of $\mathbf{x}$.

4 Convergence Analysis of Algorithm 1 Over Static Graphs

In this section, we examine the convergence behavior of Algorithm 1 over static graphs. If ρ^k is diminishing, all agents converge to the same optimal solution of problem (2) under Algorithm 1. With a constant stepsize, all agents eventually converge to a neighborhood of an optimal solution, where the error size is proportional to the stepsize. For both cases, we perform the non-asymptotic analysis to quantify their convergence rates.

Before providing the convergence results of $\{\mathbf{x}^k\}$, we recall from Proposition A.4.6 in [2] a well-known result on the convergence of a sequence of vectors.

Lemma 1 ([2]) *Let $\mathscr{X}^\star$ be a nonempty subset of $\mathbb{R}^n$, and let $\{\mathbf{x}^k\} \in \mathbb{R}^n$ be a sequence satisfying for some $p > 0$ and for all k,*

$$\|x^{k+1} - x^\star\|^p \leq \|x^k - x^\star\|^p - \gamma^k \phi(x^k) + \delta^k, \ \forall \mathbf{x}^\star \in \mathscr{X}^\star,$$

where $\{\gamma^k\}$ and $\{\delta^k\}$ are nonnegative sequences satisfying

$$\sum_{k=0}^{\infty} \gamma^k = \infty, \ \sum_{k=0}^{\infty} \delta^k < \infty.$$

Suppose that $\phi(\cdot)$ is continuous, nonnegative, and satisfies $\phi(x) = 0$ if and only if $x \in \mathscr{X}^\star$. Then $\{\mathbf{x}^k\}$ converges to an optimal point in $\mathscr{X}^\star$.

The first result in this section is on the convergence of Algorithm 1 under the assumption of diminishing stepsize, which is given as follow:

Assumption 2 The sequence $\{\rho^k\}$ satisfies

$$\sum_{k=0}^{\infty} \rho^k = \infty, \text{ and } \sum_{k=0}^{\infty} (\rho^k)^2 < \infty.$$

Proof of the convergence of Algorithm 1 under Assumption 2 is now given below.

Theorem 2 (Convergence, [28]) *Suppose that the conditions in Theorem 1 and Assumption 2 hold. Let* $\{\mathbf{x}^k\}$ *be generated by Algorithm 1. Then, there exists some optimal point* $x^\star \in \mathscr{X}^\star$ *such that* $\lim_{k\to\infty} \mathbf{x}^k = x^\star \mathbb{1}$.

Proof Under Assumption 1, we have that

$$\|\nabla \widetilde{f}_\lambda(\mathbf{x})\| \leq c_a, \forall \mathbf{x} \in \mathbb{R}^n \tag{17}$$

where $c_a = \sqrt{n}(c + \lambda \|A\|_\infty)$. Since Algorithm 1 is the exact iteration of the subgradient method of problem (7), this implies that

$$\begin{aligned} &\|\mathbf{x}^{k+1} - x^\star \mathbb{1}\|^2 \\ &= \|\mathbf{x}^k - x^\star \mathbb{1}\|^2 - 2\rho^k (\mathbf{x}^k - x^\star \mathbb{1})^\mathsf{T} \nabla \widetilde{f}_\lambda(\mathbf{x}^k) + (\rho^k)^2 \|\nabla \widetilde{f}_\lambda(\mathbf{x}^k)\|^2 \\ &\leq \|\mathbf{x}^k - x^\star \mathbb{1}\|^2 - 2\rho^k (\widetilde{f}_\lambda(\mathbf{x}^k) - \widetilde{f}_\lambda(x^\star \mathbb{1})) + (\rho^k)^2 c_a^2 \\ &\leq \|\mathbf{x}^k - x^\star \mathbb{1}\|^2 - 2\rho^k (\widetilde{f}_\lambda(\mathbf{x}^k) - f^\star) + (\rho^k)^2 c_a^2, \ \forall x^\star \in \mathscr{X}^\star \end{aligned} \tag{18}$$

where the first inequality follows from (1) and (17), and the second inequality is from Theorem 1.

By virtue of Lemma 1 and Theorem 1, the result follows immediately. □

Our next result provides a non-asymptotic result to evaluate the convergence rate for $\rho^k = k^{-\alpha}, \alpha \in (0.5, 1]$. To this end, we first define

$$d(\mathbf{x}) = \min_{x^\star \in \mathscr{X}^\star} \|\mathbf{x} - x^\star \mathbb{1}\|. \tag{19}$$

Then, it follows from (8) that

$$\begin{aligned} v(\mathbf{x}^k) &= \max_i (x_i^k) - \min_i (x_i^k) \\ \bar{x}^k &= \frac{1}{n} \mathbb{1}^\mathsf{T} \mathbf{x}^k. \end{aligned}$$

Clearly, $d(\mathbf{x})$ is the distance between $\mathbf{x}$ and the set of optimal solutions, v^k is the maximum divergence between agents' states at time k, and $\bar{x}^k$ is the mean of all agents' states at time k. Intuitively, we can use the rates that $f(\bar{x}^k)$ approaches $f^\star$ and v^k reduces to 0 to evaluate the convergence rate of Algorithm 1.

Theorem 3 *Suppose that the conditions in Theorem 1 hold, and let* $\{\mathbf{x}^k\}$ *be generated by Algorithm 1. If* $\rho^k = k^{-\alpha}, \alpha \in (0.5, 1]$, *then*

$$\begin{aligned} \min_{1<k\leq \bar{k}} f(\bar{x}^k) - f^\star &\leq \frac{(2\alpha-1)d(\mathbf{x}^0)^2 + 2\alpha c_a^2}{2(2\alpha-1)s(\bar{k})}, \\ \min_{1<k\leq \bar{k}} v(\mathbf{x}^k) &\leq \frac{(2\alpha-1)d(\mathbf{x}^0)^2 + 2\alpha c_a^2}{(2\lambda a_{min}^{(l)} - cn)(2\alpha-1)s(\bar{k})}, \end{aligned} \tag{20}$$

where $\mathbf{x}^0$ *is the initial point,* $\bar{x}^k$ *and* $v(\mathbf{x}^k)$ *are defined in (8), and*

$$s(\bar{k}) = \begin{cases} \dfrac{1}{1-\alpha}(\bar{k}^{1-\alpha} - 1), & \text{if } \alpha \in (0.5, 1), \\ ln(\bar{k}), & \text{if } \alpha = 1. \end{cases}$$

Proof By Theorem 2, $\{\mathbf{x}^k\}$ is a convergent sequence. For any $x^\star \in \mathscr{X}^\star$, it follows from (18) that

$$2\rho^k(\widetilde{f}_\lambda(\mathbf{x}^k) - f^\star) \leq \|\mathbf{x}^k - x^\star \mathbb{1}\|^2 - \|\mathbf{x}^{k+1} - x^\star \mathbb{1}\|^2 + (\rho^k)^2 c_a^2.$$

Summing the above relation over $k \in \{1, ..., \bar{k}\}$ yields

$$\begin{aligned} 2\sum_{k=1}^{\bar{k}} \rho^k(\widetilde{f}_\lambda(\mathbf{x}^k) - f^\star) &\leq \|\mathbf{x}^0 - x^\star \mathbb{1}\|^2 - \|\mathbf{x}^{\bar{k}+1} - x^\star \mathbb{1}\|^2 + \sum_{k=1}^{\bar{k}} (\rho^k)^2 c_a^2 \\ &\leq d(\mathbf{x}^0)^2 + \sum_{k=1}^{\bar{k}} (\rho^k)^2 c_a^2 \end{aligned}$$

where the last inequality holds by choosing $x^\star = \text{argmin}_{x\in\mathscr{X}^\star}\|\mathbf{x}^0 - x\mathbb{1}\|$. Then, we arrive at

$$\min_{0\leq k\leq \bar{k}} \widetilde{f}_\lambda(\mathbf{x}^k) - f^\star \leq \frac{d(\mathbf{x}^0)^2 + \sum_{k=1}^{\bar{k}} (\rho^k)^2 c_a^2}{2\sum_{k=1}^{\bar{k}} \rho^k}. \tag{21}$$

Since $\int_1^{\bar{k}} x^{-\alpha} dx < \sum_{k=1}^{\bar{k}} k^{-\alpha} < \int_1^{\bar{k}} x^{-\alpha} dx + 1$, we have that

$$\sum_{k=1}^{\bar{k}} (\rho^k)^2 < \int_1^{\bar{k}} x^{-2\alpha} dx + 1 = \frac{1 - \bar{k}^{1-2\alpha}}{2\alpha - 1} + 1 < \frac{2\alpha}{2\alpha - 1},$$

and $\sum_{k=1}^{\bar{k}} \rho^k > \int_1^{\bar{k}} x^{-\alpha} dx = s(\bar{k})$. Using the above and (21) leads to

$$\min_{0\le k\le \bar{k}} \widetilde{f}_\lambda(\mathbf{x}^k) - f^\star \le \frac{(2\alpha-1)d(\mathbf{x}^0)^2 + 2\alpha c_a^2}{2(2\alpha-1)s(\bar{k})}. \tag{22}$$

Since $f(\bar{x}^k) - f^\star > 0$ and $\lambda a_{\min}^{(l)} - \frac{1}{2}cn > 0$, it follows from (15) and (22) that (20) holds. □

The first inequality in (20) quantifies the decreasing rate of the gap between $f(\bar{x}^k)$ and the optimal value $f^\star$, while the second one shows that the largest difference between agents' states is reduced at a comparable rate. Thus, Theorem 3 reveals that the convergence rate lies between $O(1/\ln(k))$ and $O(\ln(k)/\sqrt{k})$, depending on the choice of ρ^k.

We also provide an alternative evaluation of the convergence rate, which uses a robust form and is presented in the following Corollary 1.

Corollary 1 (Non-asymptotic convergence, [28]) *Suppose that the conditions in Theorem 3 hold. Then*

$$\min_{1<k\le \bar{k}} \max_{i\in\mathcal{V}} f(x_i^k) - f^\star \le \frac{(2\alpha-1)d(\mathbf{x}^0)^2 + 2\alpha c_a^2}{2(2\alpha-1)s(\bar{k})}$$

where all notations are the same as those in Theorem 3.

Proof For all k and any $x_m \in [\min_{i\in\mathcal{V}} x_i^k, \max_{i\in\mathcal{V}} x_i^k]$, it follows from (11) that

$$f(x_m) \le \widetilde{f}_\lambda(\mathbf{x}^k) - \lambda h(\mathbf{x}^k) + \|\mathbf{x}^k - x_m\mathbb{1}\|\|\nabla g(x_m\mathbb{1})\|$$

which together with

$$\|\mathbf{x}^k - x_m\mathbb{1}\|\|\nabla g(x_m\mathbb{1})\| \le \sqrt{n}\|\mathbf{x}^k - x_m\mathbb{1}\|_\infty \cdot \sqrt{n}\|\nabla g(x_m\mathbb{1})\|_\infty \le ncv(\mathbf{x}^k)$$

and (13) yields that

$$\begin{aligned} f(x_m) &\le \widetilde{f}_\lambda(\mathbf{x}^k) - \lambda h(\mathbf{x}^k) + \frac{nc}{a_{\min}^{(l)}} h(\mathbf{x}^k) \\ &= g(\mathbf{x}^k) + \frac{nc}{a_{\min}^{(l)}} h(\mathbf{x}^k) \\ &\le \widetilde{f}_{2\lambda}(\mathbf{x}^k) \end{aligned}$$

where the last inequality follows from $\lambda > nc/(2a_{\min}^{(l)})$.

Noting that (22) implies

$$\min_{0\le k\le \bar{k}} \widetilde{f}_{2\lambda}(\mathbf{x}^k) - f^\star \le \frac{(2\alpha-1)d(\mathbf{x}^0)^2 + 2\alpha c_a^2}{2(2\alpha-1)s(\bar{k})}$$

the result follows immediately. □

If $f(x)$ is non-differentiable, the objective function of the classical distributed algorithm (4) converges at a rate of $O(\ln(k)/\sqrt{k})$ when $\rho^k = 1/\sqrt{k}$ [20], which is comparable to Algorithm 1 when α approaches 0.5. Thus using only the sign of relative state essentially does not lead to any reduction in the convergence rate. However, if $f(x)$ is more smooth, e.g., differentiable or strongly convex, Algorithm 1 may converge at a rate slower than that of (4).

For a constant stepsize, Algorithm 1 approaches a neighborhood of an optimal solution as fast as $O(1/k)$ and the error size is proportional to the stepsize. These are formally stated in Theorems 4 and 5.

Theorem 4 (Constant Stepsize, [28]) *Suppose that the conditions in Theorem 1 hold, and let* $\{\mathbf{x}^k\}$ *be generated by Algorithm 1. If* $\rho^k = \rho$, *then*

$$\limsup_{k\to\infty} d(\mathbf{x}^k) \le 2\sqrt{n}\max\left\{\widetilde{d}(\rho), \frac{\rho c_a^2}{2\lambda a_{min}^{(l)} - cn}\right\} + \rho c_a$$

where $\widetilde{\mathscr{X}}(\rho) = \{x | f(x) \le f^\star + \rho c_a^2/2\}$ *and* $\widetilde{d}(\rho) = \max_{x\in\widetilde{\mathscr{X}}(\rho)} d(x) < \infty$.

Proof See the Appendix. □

In Theorem 4, $\widetilde{d}(0) = 0$ and $\widetilde{d}(\rho)$ is increasing in ρ. Thus, Algorithm 1 under a constant stepsize finally approaches a neighborhood of $x^\star\mathbb{1}$ for some $x^\star \in \mathscr{X}^\star$, the size of which decreases to zero as ρ tends to zero. If the order of growth of f near the set of optimal solutions is available, then $\widetilde{d}(\rho)$ can even be determined explicitly, which is illustrated in Corollary 2.

Corollary 2 ([28]) *Suppose that the conditions in Theorem 4 hold, and that* $f(x)$ *satisfies*

$$f(x) - f^\star \ge \gamma(d(x))^\alpha$$

where $\gamma > 0$ *and* $\alpha \ge 1$. *Then, it holds that*

$$\limsup_{k\to\infty} d(\mathbf{x}^k) \le 2\sqrt{n}\max\left\{\left(\frac{\rho c_a^2}{2\gamma}\right)^{\frac{1}{\alpha}}, \frac{\rho c_a^2}{2\lambda a_{min}^{(l)} - cn}\right\} + \rho c_a$$

Proof Noting that $\widetilde{d}(\rho) \le (\rho c_a^2/2\gamma)^{\frac{1}{\alpha}}$, the result follows directly from Theorem 4. □

The following theorem evaluates the convergence rate when the stepsize is set to be constant.

Theorem 5 ([28]) *Suppose that the conditions in Theorem 4 hold. Then*

$$\min_{0\le k\le \bar{k}} f(\bar{x}^k) - f^\star \le \frac{\rho c_a^2}{2} + \frac{d(\mathbf{x}^0)^2}{2\rho\bar{k}}, \tag{23}$$

$$\min_{0\le k\le \bar{k}} v(\mathbf{x}^k) \le \frac{\rho c_a^2}{2\lambda a_{min}^{(l)} - cn} + \frac{d(\mathbf{x}^0)^2}{\rho\bar{k}(2\lambda a_{min}^{(l)} - cn)}.$$

Proof From (21) we know that

$$\min_{0\le k\le \bar{k}} \widetilde{f}_\lambda(\mathbf{x}^k) - f^\star \le \frac{d(\mathbf{x}^0)^2 + \bar{k}\rho^2 c_a^2}{2\rho\bar{k}},$$

which together with (15) implies the result. □

Remark 3 The following conclusions can be easily drawn from Theorem 5.

(a) $\min_{0\le k\le \bar{k}} f(\bar{x}^k)$ approaches the interval $[f^\star, f^\star + \frac{\rho c_a^2}{2}]$ at a rate of $O(1/\bar{k})$.

(b) Given $\bar{k}$ iterations, let $\rho = \frac{1}{c_a}\frac{d(\mathbf{x}^0)}{\sqrt{\bar{k}}}$, which minimizes the right-hand side of (23). Then

$$\min_{0\le k\le \bar{k}} f(\bar{x}^k) - f^\star \le c_a \frac{d(\mathbf{x}^0)}{\sqrt{\bar{k}}},$$

$$\min_{0\le k\le \bar{k}} v(\mathbf{x}^k) \le \frac{c_a}{2\lambda a_{\min}^{(l)} - cn}\frac{d(\mathbf{x}^0)}{\sqrt{\bar{k}}}.$$

The multi-agent network converges only to a point that is close to an optimal solution with an error size $O(\bar{k}^{-1/2})$.

Algorithm 2: Distributed Algorithm Using the Sign of Relative State

1. **Initialization**: Every node i sets $x_i^0 = 0$ for all $i \in \mathcal{V}$.
2. **Repeat**
3. **Information collection**: Each node i collects the sign of the relative state to its neighbors at time k, e.g., node $j \in \mathcal{N}_i^k$ and obtain r_i^k, which is given below

$$r_i^k = \sum_{j\in\mathcal{N}_i^k} a_{ij}^k \operatorname{sgn}(x_j^k - x_i^k).$$

4. **Local update**: The decision variable in each node is locally updated as

$$x_i^{k+1} = x_i^k + \rho^k\left(\lambda \cdot r_i^k - \nabla f_i(x_i^k)\right),$$

where λ and ρ^k are to be given, and $\nabla f_i(x_i^k)$ is any subgradient of $f_i(x)$ at x_i^k.
5. **Set** $k = k + 1$.
6. **Until** a predefined stopping rule (e.g., a maximum iteration number) is satisfied.

5 The Distributed Optimization Algorithm over Time-varying Graphs

When the graphs are time-varying, Algorithm 1 is revised and we provide the details in Algorithm 2. In this section, we study the convergence of Algorithm 2 over two types of time-varying graphs: uniformly jointly connected time-varying graphs and *randomly* activated graphs.

5.1 Uniformly Jointly Connected Time-varying Graphs

Now we introduce the concept of uniformly jointly connected time-varying graphs. First we define the union of the graphs $\mathcal{G}^{(k,b)}$ for integers $k \geq 0$ and $b > 0$ below

$$\mathcal{G}^{(k,b)} = (\mathcal{V}, \mathcal{E}^{(k,b)}, A^{(k,b)}) := \mathcal{G}^k \cup \mathcal{G}^{k+1} \cup \cdots \cup \mathcal{G}^{k+b-1}$$

and $A^{(k,b)}$ is the associated adjacency matrix of $\mathcal{G}^{(k,b)}$. We make the following assumption.

Assumption 3 Assume that

(a) For some $\eta > 0$, it holds that

$$\begin{cases} a_{ij}^k \geq \eta, \text{ if } (i, j) \in \mathcal{E}^k, \\ a_{ij}^k = 0, \text{ otherwise.} \end{cases} \tag{24}$$

(b) There exists an integer $b \geq 1$ such that $A^{(tb,b)}$ is l-connected for each $t = 0, 1, 2, \ldots$

Assumption 3 is commonly made in dealing with deterministically time-varying graphs. The first part requires that either an edge is not connected at some time, or the edge is connected with a weight larger than some fixed value. The second part assumes the joint graph in time intervals with length b to be connected. We call time-varying graphs satisfying Assumption 3 *uniformly jointly connected graphs*, which are also sometimes referred to as b-connected graphs [15, 17].

We are now ready to present the convergence result of Algorithm 2 over uniformly jointly connected graphs.

Theorem 6 (Convergence, [26]) *Suppose that Assumptions 1-3 hold, and that there exists a constant $c_\rho > 0$ such that for all $k > 0$,*

$$\max_{t \in [k, k+b)} \rho^t \leq c_\rho \min_{t \in [k, k+b)} \rho^t. \tag{25}$$

Select

$$\lambda > \frac{nbcc_\rho}{2l\eta},$$

where n is the number of agents, c is given in Assumption 1, c_ρ is given in Assumption 2, and b, l, η are given in Assumption 3. Let $\{\mathbf{x}^k\}$ be generated by Algorithm 2. Then, $\lim_{k\to\infty} \mathbf{x}^k = x^\star \mathbb{1}$ for some $x^\star \in \mathscr{X}^\star$.

Proof We first consider the subsequence $\{\mathbf{x}^{tb}, t = 0, 1, 2, ...\}$, i.e., we let $k = tb, t \in \{0, 1, 2, ...\}$. Define

$$\widetilde{f}_\lambda^k(\mathbf{x}) := \frac{\lambda}{2} \sum_{i,j\in\mathscr{V}} a_{ij}^k |x_i - x_j| + \sum_{i=1}^n f_i(x_i)$$

and

$$\begin{aligned}
\widetilde{f}_\lambda^{(k,b)}(\mathbf{x}) &:= \frac{1}{\rho^k} \sum_{t=k}^{b+k-1} \rho^t \widetilde{f}_\lambda^t(\mathbf{x}) \\
&= \frac{\lambda}{2\rho^k} \sum_{i,j\in\mathscr{V}} \sum_{t=k}^{b+k-1} \rho^t a_{ij}^t |x_i - x_j| + \frac{1}{\rho^k} \sum_{t=k}^{b+k-1} \rho^t \sum_{i=1}^n f_i(x_i) \\
&= \bar{\rho}^k \left[\frac{\lambda}{2} \sum_{i,j\in\mathscr{V}} \bar{a}_{ij}^k |x_i - x_j| + \sum_{i=1}^n f_i(x_i) \right]
\end{aligned}$$

where

$$\bar{\rho}^k = \sum_{t=k}^{b+k-1} \frac{\rho^t}{\rho^k}, \text{ and } \bar{a}_{ij}^k = \frac{\sum_{t=k}^{b+k-1} \rho^t a_{ij}^t}{\sum_{t=k}^{b+k-1} \rho^t}.$$

Let $[\bar{A}^k]_{ij} := \bar{a}_{ij}^k$ and $\bar{a}_{\min}^{k,(l)}$ be the sum of the l smallest nonzero elements of $\bar{A}^k$. Note that $\bar{a}_{\min}^{k,(l)}$ is well defined because for any (i, j), if $[A^{(k,b)}]_{ij}$ is nonzero, then $[\bar{A}^k]_{ij}$ is also nonzero, and $A^{(k,b)}$ has at least l nonzero elements by Assumption 3.

Then, we obtain from (25) that

$$\bar{a}_{ij}^k \geq \frac{\min_{t\in[k,k+b)} \rho^t \sum_{t=k}^{b+k-1} a_{ij}^t}{b \max_{t\in[k,k+b)} \rho^t} \geq \frac{\sum_{t=k}^{b+k-1} a_{ij}^t}{bc_\rho}.$$

Thus, if $\bar{a}_{ij}^k \neq 0$, then it follows from (24) that $\bar{a}_{ij}^k$ must be larger than η/bc_ρ, which means that any nonzero element of $\bar{A}^k$ is larger than η/bc_ρ, and hence $\bar{a}_{\min}^{k,(l)} \geq l\eta/bc_\rho$.

By virtue of that $\lambda > nbcc_\rho/(2l\eta)$ and Theorem 1, we know that the problem

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}}\ \frac{1}{\bar{\rho}^k}\widetilde{f}_\lambda^{(k,b)}(\mathbf{x})$$

is equivalent to the original problem for all $k = tb, t \in \{0, 1, 2, ...\}$. That is, we have $\widetilde{f}_\lambda^{(k,b)}(\mathbf{x}) \geq \bar{\rho}^k f^\star$ for all $\mathbf{x} \in \mathbb{R}^n$, and $\widetilde{f}_\lambda^{(k,b)}(\mathbf{x}) = \bar{\rho}^k f^\star$ if and only if $\mathbf{x} \in \{a\mathbb{1}|a \in \mathscr{X}^\star\}$.

Let $\mathbf{d}^k = [d_1^k, ..., d_n^k]^\mathsf{T}$, where

$$d_i^k = -\lambda \sum_{j\in\mathscr{N}_i^k} a_{ij}^k \mathrm{sgn}(x_j^k - x_i^k) + \nabla f_i(x_i^k).$$

Then, Algorithm 2 can be written in a compact form as

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \rho^k \mathbf{d}^k.$$

Note that $\mathbf{d}^k$ is a subgradient of $\widetilde{f}_\lambda^k(\mathbf{x})$ at $\mathbf{x}^k$, and $\|\nabla \widetilde{f}_\lambda^k(\mathbf{x})\| \leq c_a$ for any $\mathbf{x} \in \mathbb{R}^n$ by (17). Hence $\|\mathbf{d}^k\| \leq c_a$ for any k. Let $x^\star$ be an arbitrary element of $\mathscr{X}^\star$. We have the following relation

$$\begin{aligned}\|\mathbf{x}^{k+b} - x^\star\mathbb{1}\|^2 &= \left\|\mathbf{x}^k - \sum_{t=k}^{b+k-1}\rho^t\mathbf{d}^t - x^\star\mathbb{1}\right\|^2 \\ &= \|\mathbf{x}^k - x^\star\mathbb{1}\|^2 + 2(x^\star\mathbb{1} - \mathbf{x}^k)^\mathsf{T}\sum_{t=k}^{b+k-1}\rho^t\mathbf{d}^t + \left\|\sum_{t=k}^{b+k-1}\rho^t\mathbf{d}^t\right\|^2 \\ &\leq \|\mathbf{x}^k - x^\star\mathbb{1}\|^2 + 2(x^\star\mathbb{1} - \mathbf{x}^k)^\mathsf{T}\sum_{t=k}^{b+k-1}\rho^t\mathbf{d}^t + bc_a^2\sum_{t=k}^{b+k-1}(\rho^t)^2.\end{aligned} \tag{26}$$

Consider the second term of the right-hand-side of (26); then

$$\begin{aligned}&(x^\star\mathbb{1} - \mathbf{x}^k)^\mathsf{T}\sum_{t=k}^{b+k-1}\rho^t\mathbf{d}^t = \sum_{t=k}^{b+k-1}\rho^t(x^\star\mathbb{1} - \mathbf{x}^k)^\mathsf{T}\mathbf{d}^t \\ &= \sum_{t=k}^{b+k-1}\rho^t(x^\star\mathbb{1} - \mathbf{x}^t)^\mathsf{T}\mathbf{d}^t + \sum_{t=k}^{b+k-1}\rho^t(\mathbf{x}^t - \mathbf{x}^k)^\mathsf{T}\mathbf{d}^t \\ &\leq \sum_{t=k}^{b+k-1}\rho^t(\widetilde{f}_\lambda^t(x^\star\mathbb{1}) - \widetilde{f}_\lambda^t(\mathbf{x}^t)) + \sum_{t=k}^{b+k-1}\rho^t(\mathbf{x}^t - \mathbf{x}^k)^\mathsf{T}\mathbf{d}^t \\ &= \sum_{t=k}^{b+k-1}\rho^t(f^\star - \widetilde{f}_\lambda^t(\mathbf{x}^k)) + \sum_{t=k}^{b+k-1}\rho^t(\widetilde{f}_\lambda^t(\mathbf{x}^k) - \widetilde{f}_\lambda^t(\mathbf{x}^t)) + \sum_{t=k}^{b+k-1}\rho^t(\mathbf{x}^t - \mathbf{x}^k)^\mathsf{T}\mathbf{d}^t\end{aligned} \tag{27}$$

$$
\begin{aligned}
&= \rho^k(f^\star - \widetilde{f}_\lambda^{(k,b)}(\mathbf{x}^k)) + \sum_{t=k}^{b+k-1} \rho^t(\widetilde{f}_\lambda^t(\mathbf{x}^k) - \widetilde{f}_\lambda^t(\mathbf{x}^t)) + \sum_{t=k}^{b+k-1} \rho^t(\mathbf{x}^t - \mathbf{x}^k)^\mathsf{T}\mathbf{d}^t \\
&\le \rho^k(f^\star - \widetilde{f}_\lambda^{(k,b)}(\mathbf{x}^k)) + \sum_{t=k}^{b+k-1} \rho^t \left(\|\mathbf{x}^t - \mathbf{x}^k\|\|\mathbf{d}^t\| + \|\mathbf{x}^k - \mathbf{x}^t\|\|\nabla \widetilde{f}_\lambda^t(\mathbf{x}^k)\|\right) \\
&= \rho^k(f^\star - \widetilde{f}_\lambda^{(k,b)}(\mathbf{x}^k)) + \sum_{t=k}^{b+k-1} \rho^t \|\mathbf{x}^k - \sum_{u=k}^{t-1} \rho^u \mathbf{d}^u - \mathbf{x}^k\|(\|\mathbf{d}^t\| + \|\nabla \widetilde{f}_\lambda^t(\mathbf{x}^k)\|) \\
&\le \rho^k(f^\star - \widetilde{f}_\lambda^{(k,b)}(\mathbf{x}^k)) + (\sum_{t=k}^{b+k-1} \rho^t)^2 \\
&\le \rho^k(f^\star - \widetilde{f}_\lambda^{(k,b)}(\mathbf{x}^k)) + 2bc_a^2 \sum_{t=k}^{b+k-1} (\rho^t)^2.
\end{aligned}
$$

Combining (27) with (26) yields that

$$
\|\mathbf{x}^{k+b} - x^\star \mathbb{1}\|^2 \le \|\mathbf{x}^k - x^\star \mathbb{1}\|^2 + 2\rho^k(f^\star - \widetilde{f}_\lambda^{(k,b)}(\mathbf{x}^k)) + 5bc_a^2 \sum_{t=k}^{b+k-1} (\rho^t)^2. \quad (28)
$$

Noting that $k = tb, t \in \{0, 1, ...\}$, the above relation becomes

$$
\begin{aligned}
&\|\mathbf{x}^{(t+1)b} - x^\star \mathbb{1}\|^2 \\
&\le \|\mathbf{x}^{tb} - x^\star \mathbb{1}\|^2 + 2\rho^{tb}(f^\star - \widetilde{f}_\lambda^{(tb,b)}(\mathbf{x}^k)) + 5bc_a^2 \sum_{u=tb}^{(t+1)b-1} (\rho^u)^2.
\end{aligned}
$$

Note that $\widetilde{f}_\lambda^{(tb,b)}(\mathbf{x}^k)$ is nonnegative and $\widetilde{f}_\lambda^{(tb,b)}(\mathbf{x}) = 0$ if and only if $\mathbf{x} \in \{a\mathbb{1} | a \in \mathscr{X}^\star\}$, and that $\sum_{t=1}^\infty \rho^{tb} = \infty, \sum_{t=1}^\infty (\rho^{tb})^2 < \infty$. It follows from Lemma 1 that there exists $\bar{x} \in \mathscr{X}^\star$ such that the subsequence $\{\mathbf{x}^{tb}\}, t \in \{0, 1, 2, ...\}$ must converge to $\bar{x}\mathbb{1}$. This, combined with $\lim_{k\to\infty} \rho^k = 0$, implies that $\{\mathbf{x}^k\}$ converges to $\bar{x}\mathbb{1}$. □

Compared with the convergence result on static graphs (Theorem 2), the major difference on uniformly jointly connected graphs is that λ should be bc_ρ times larger than that in the case of static graphs.

Next, we evaluate the convergence rate of Algorithm 2 over uniformly jointly connected graphs when $\rho^k = k^{-\alpha}, \alpha \in (0.5, 1]$. As in Theorem 3, we evaluate the rates that $f(\bar{x}^k)$ approaches $f^\star$ and $v(\mathbf{x}^k)$ tends to 0 to quantify the convergence rate.

Theorem 7 (Non-asymptotic result, [26]) *Let the assumptions in Theorem 6 hold, and further assume that $\lambda > nbc/l\eta$. Let $\{\mathbf{x}^k\}$ be generated by Algorithm 2. If $\rho^k = k^{-\alpha}$ with some $\alpha \in (0.5, 1]$, then for any $k_0 > 2b$,*

$$\min_{1<k\le k_0} f(\bar{x}^k) - f^\star \le \frac{(2\alpha-1)(d(\mathbf{x}^0))^2 + 10\alpha b c_a^2}{b(2\alpha-1)s(k_0)} \tag{29}$$
$$\min_{1<k\le k_0} v(\mathbf{x}^k) \le \frac{2(2\alpha-1)(d(\mathbf{x}^0))^2 + 12\alpha b c_a^2}{(\lambda l\eta - nbc)(2\alpha-1)s(k_0)}$$

where $\mathbf{x}^0$ *is the initial point, and*

$$s(k_0) = \begin{cases} \dfrac{(k_0-b)^{1-\alpha} - b^{1-\alpha}}{b(1-\alpha)}, & \alpha \in (0.5, 1), \\ \dfrac{1}{b}[ln(k_0-b) - ln(b)], & \alpha = 1. \end{cases}$$

Proof Note that λ and $\{\rho^k\}$ satisfy the conditions in Theorem 6 with $c_\rho = 2$, and $\|\nabla \widetilde{f}_\lambda^k(\mathbf{x})\| \le c_a$ for any $\mathbf{x}$ and k. Let $x^\star$ be an arbitrary optimal solution of problem (2) and $t_0 = \lfloor k_0/b \rfloor$, where $\lfloor x \rfloor$ denotes the nearest integer to $(\cdot)$ that is smaller than $(\cdot)$. It then follows from (28) that

$$2\rho^{tb}(\widetilde{f}_\lambda^{(tb,b)}(\mathbf{x}^k) - f^\star) \le \|\mathbf{x}^{tb} - x^\star \mathbb{1}\|^2 - \|\mathbf{x}^{(t+1)b} - x^\star \mathbb{1}\|^2 + 5bc_a^2 \sum_{u=tb}^{tb+b-1} (\rho^u)^2.$$

Summing the above relation over $t = 0, 1, ..., t_0$ yields

$$\begin{aligned} &2\sum_{t=0}^{t_0} \rho^{tb}(\widetilde{f}_\lambda^{(tb,b)}(\mathbf{x}^k) - f^\star) \\ &\le \|\mathbf{x}^0 - x^\star \mathbb{1}\|^2 - \|\mathbf{x}^{t_0 b+1} - x^\star \mathbb{1}\|^2 + 5bc_a^2 \sum_{t=0}^{t_0} \sum_{u=tb}^{(t+1)b-1} (\rho^u)^2 \\ &\le d(\mathbf{x}^0) + 5bc_a^2 \sum_{k=1}^{k_0} (\rho^k)^2. \end{aligned}$$

Therefore, we have

$$\min_{0\le k\le k_0} \widetilde{f}_\lambda^{(k,b)}(\mathbf{x}^k) - f^\star \le \frac{d(\mathbf{x}^0) + 5bc_a^2 \sum_{k=1}^{k_0} (\rho^k)^2}{2\sum_{t=0}^{t_0} \rho^{tb}}. \tag{30}$$

Since

$$\int_1^{k_0} \frac{1}{x^\alpha} dx < \sum_{k=1}^{k_0} \frac{1}{k^\alpha} < \int_1^{k_0} \frac{1}{x^\alpha} dx + 1,$$

we obtain that

$$\sum_{k=1}^{k_0}(\rho^k)^2 < \int_1^{k_0}\frac{1}{x^{2\alpha}}dx + 1 = \frac{1-k_0^{1-2\alpha}}{2\alpha-1} + 1 < \frac{2\alpha}{2\alpha-1}$$

and for $\alpha \in (0.5, 1)$,

$$\begin{aligned}\sum_{t=0}^{t_0}\rho^{tb} &> b^{-a}\sum_{t=0}^{t_0}\rho^t > b^{-a}\int_1^{t_0}\frac{1}{x^\alpha}dx = \frac{t_0^{1-\alpha}-1}{b^a(1-\alpha)}\\ &> \frac{(k_0/b-1)^{1-\alpha}-1}{b^a(1-\alpha)} = s(k_0).\end{aligned}$$

We also obtain $\sum_{t=0}^{t_0}\rho^{tb} = s(k_0)$ using similar arguments. Substituting these two inequalities into (30) yields

$$\min_{0\le k\le k_0}\widetilde{f}_\lambda^{(k,b)}(\mathbf{x}^k) - f^\star \le \frac{(2\alpha-1)d(\mathbf{x}^0) + 10bc_a\alpha}{2(2\alpha-1)s(k_0)}. \tag{31}$$

Noticing that $\bar{\rho}^k \ge c_\rho/2 \ge b/2$ for all k, we have

$$\begin{aligned}\widetilde{f}_\lambda^{(k,b)}(\mathbf{x}^k) &= \lambda\bar{\rho}^k h(\mathbf{x}^k) + \bar{\rho}^k g(\mathbf{x}^k - \bar{x}^k\mathbb{1} + \bar{x}^k\mathbb{1})\\ &\ge \frac{b}{2}\left[\lambda h(\mathbf{x}^k) + g(\bar{x}^k\mathbb{1}) + (\mathbf{x}^k - \bar{x}^k\mathbb{1})^{\mathsf{T}}\nabla g(\bar{x}^k\mathbb{1})\right]\\ &\ge \frac{b}{2}\left[\lambda h(\mathbf{x}^k) + f(\bar{x}^k) - \|\mathbf{x}^k - \bar{x}^k\mathbb{1}\|\|\nabla g(\bar{x}^k\mathbb{1})\|\right]\end{aligned} \tag{32}$$

where the first equality follows from the definition of $\widetilde{f}_\lambda^{(k,b)}(\mathbf{x})$, the second inequality is from the definition of a subgradient, and the last inequality is the result of the Cauchy–Schwarz inequality as well as the fact that $g(a\mathbb{1}) = f(a)$.

Recall from (13) and (14) that

$$h(\mathbf{x}^k) \ge \frac{l\eta}{2b}v^k, \text{ and } \|\mathbf{x}^k - \bar{x}^k\mathbb{1}\|\|\nabla g(\bar{x}^k\mathbb{1})\| \le \frac{nc}{2}v(\mathbf{x}^k).$$

These two relations together with (32) yield

$$\widetilde{f}_\lambda^{(k,b)}(\mathbf{x}^k) - f^\star \ge \frac{b}{2}\left[f(\bar{x}^k) - f^\star + (\frac{\lambda l\eta}{2b} - \frac{nc}{2})v(\mathbf{x}^k)\right].$$

Since $f(\bar{x}^k) - f^\star > 0$ and $\lambda l\eta - bcn > 0$, the above inequality combined with (31) implies (29) immediately. □

Theorem 2 reveals that from the worst-case point of view, the convergence rate over uniformly jointly connected time-varying graphs is about b times slower than that of a static graph (Theorem 3), which is reasonable.

5.2 Randomly Activated Graphs

This subsection studies the convergence of Algorithm 2 over randomly activated graphs, which can model many networks such as gossip social networks and random measurement losses in networks. The definition is given as follows.

Definition 2 (*Randomly Activated Graphs*) The sequence of graphs $\{\mathcal{G}^k\}$ are randomly activated if for all $i, j \in \mathcal{V}, i \neq j$, $\{a_{ij}^k\}$ is an i.i.d. Bernoulli process with $\mathbb{P}\{a_{ij}^k = 1\} = p_{ij}$, where $\mathbb{P}(\mathcal{X})$ denotes the probability of an event $\mathcal{X}$ and $0 \leq p_{ij} \leq 1,\ \forall i, j \in \mathcal{V}$.

Remark 4 For brevity, we assume here that the weight of each edge a_{ij}^k is taken to be either zero or one at each time k in randomly activated graphs.

We call $P = [p_{ij}]$ the activation matrix of $\mathcal{G}^k$, and the graph associated with P is denoted as $\mathcal{G}_P$, which is also the mean graph of $\mathcal{G}^k$, i.e.,

$$\mathcal{G}_P := \mathbb{E}(\mathcal{G}^k). \tag{34}$$

Recall that Algorithm 1 is the iteration of subgradient methods of (7). Similarly, Algorithm 2 is just the iteration of the *stochastic* subgradient method of the following optimization problem

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}}\ \widehat{f}_\lambda(\mathbf{x}) := g(\mathbf{x}) + \lambda\widehat{h}(\mathbf{x}) \tag{35}$$

where $g(x)$ is given in (5) and

$$\widehat{h}(\mathbf{x}) = \frac{1}{2}\sum_{i,j=1}^{n} p_{ij}|x_i - x_j|.$$

To exposit it, notice that $\mathbb{E}(a_{ij}^k) = p_{ij}$, and thus a stochastic subgradient $\nabla_s\widehat{h}(\mathbf{x}) = [\nabla_s\widehat{h}(\mathbf{x})_1, ..., \nabla_s\widehat{h}(\mathbf{x})_n]^\mathsf{T}$ of $\widehat{h}(\mathbf{x})$ is given element-wise by

$$\nabla_s\widehat{h}(\mathbf{x})_i = \sum_{j=1}^{n} a_{ij}^k \text{sgn}(x_i - x_j) = \sum_{j\in\mathcal{N}_i^k} \text{sgn}(x_i - x_j).$$

Since $\mathbb{E}\{\nabla_s\widehat{h}(\mathbf{x})_i\} = \sum_j p_{ij}\text{sgn}(x_i - x_j)$, $\mathbb{E}\{\nabla_s\widehat{h}(\mathbf{x})\}$ is a subgradient of $\widehat{h}(\mathbf{x})$. Hence, the almost sure convergence of Algorithm 2 follows from the following lemma.

Lemma 2 (Convergence of Stochastic Subgradient Method, [3]) *Consider the optimization problem*

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\textit{minimize}}\ \mathbb{E}\{F(\mathbf{x}, w)\} \tag{36}$$

where w is a random variable and $F(\mathbf{x}, w) : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$ is continuous and convex w.r.t. $\mathbf{x}$ for any given w. Let $\mathscr{X}^\star$ be the set of optimal solutions and assume that $\mathscr{X}^\star$ is not empty.

The stochastic subgradient method for (36) is given by

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \rho^k r(\mathbf{x}^k, w^k)$$

where $r(\mathbf{x}, w^k)$ is bounded and $\mathbb{E}(r(\mathbf{x}, w^k))$ is a subgradient of $\mathbb{E}\{F(\mathbf{x}, w^k)\}$ for all $\mathbf{x} \in \mathbb{R}^n$. If $\{\rho^k\}$ is chosen such that

$$\sum_{k=0}^{\infty} \rho^k = \infty, \quad \sum_{k=0}^{\infty} (\rho^k)^2 < \infty,$$

then it holds almost surely that $\lim_{k\to\infty} \mathbf{x}^k = \mathbf{x}^\star$ for some $\mathbf{x}^\star \in \mathscr{X}^\star$.

The following theorem summarizes the above analysis, and is the main result of this subsection.

Theorem 8 ([28]) *Suppose that Assumptions 1 and 2 hold, and that the multi-agent network $\mathscr{G}_P$ is l-connected. Select*

$$\lambda > \frac{nc}{2p_{\min}^{(l)}},$$

where $\mathscr{G}_P$ is given in (34), $p_{\min}^{(l)}$ denotes the sum of the l smallest nonzero elements of P. Let $\{\mathbf{x}^k\}$ be generated by Algorithm 2. Then, it holds almost surely that $\lim_{k\to\infty} \mathbf{x}^k = x^\star \mathbb{1}$ for some $x^\star \in \mathscr{X}^\star$.

Proof By Theorem 1, it follows that problem (35) has the same set of optimal solutions and optimal value as problem (2). Combined with Lemma 2, the proof follows. □

6 Numerical Examples

In this section, we apply our algorithms to distributedly find the geometric median of a couple of points in a two-dimensional plane. The geometric median of n points is defined as the point which minimizes the sum of Euclidean distances to these points [7]. In other words, it is the optimal solution of the following convex optimization problem:

$$\underset{\mathbf{x}\in\mathbb{R}^2}{\text{minimize}} \ f(\mathbf{x}) := \sum_{i=1}^{n} f_i(\mathbf{x}) = \sum_{i=1}^{n} \|\mathbf{x} - \mathbf{x}_i\|. \tag{37}$$

The local function $f_i(\mathbf{x}) := \|\mathbf{x} - \mathbf{x}_i\|$ is convex but non-differentiable, the subdifferential of which is given as

$$\partial f_i(\mathbf{x}) = \begin{cases} \dfrac{\mathbf{x} - \mathbf{x}_i}{\|\mathbf{x} - \mathbf{x}_i\|}, & \text{if } \mathbf{x} \neq \mathbf{x}_i \\ \{g \mid \|g\| \leq 1\}, & \text{otherwise.} \end{cases}$$

Apparently, problem (37) satisfies Assumption 1, and hence it can be solved by Algorithms 1 and 2. Notice that $\mathbf{x}$ in (37) is 2-dimensional and Algorithms 1 and 2 should be modified accordingly as stated in Remark 1.

The geometric median problem is a special case of least square problems in statistics and Weber problems in location theory. Here we provide a possible application in distributed settings. Consider n base stations under the sea, and we want to find a place to build a communication center, which should have the minimum distances to these stations to save the costs of cables. Since global positioning is very difficult under seas, a feasible distributed approach to find the desired place is for each station to send an agent, which however can only measure the distance to the station and know rough relative positions to its neighbor agents. Clearly, we can use the proposed algorithms to achieve this goal.

In this example, we consider five stations (hence five agents), the positions of which are randomly generated on a rectangular area with size 100×100. We run three simulations over a static graph, uniformly jointly connected graphs, and randomly activated graphs, respectively. We choose the stepsize $\rho^k = 5/(k + 10)$ in all simulations. The topology of the five agents is a ring graph as shown in Fig. 2a. The λ in Algorithm 1 used in the static graph's case is chosen to be 2, which satisfies the condition in Remark 2. For the uniformly jointly connected graphs' case, we let only one edge in the graph of Fig. 2a connect at each time, and each edge connects once and only once in each cycle, the order of which is determined by a random permutation of $\{1, ..., 5\}$ at the beginning of each cycle. The λ in Algorithm 2 used in the case of uniformly jointly connected graphs is chosen to be 6. We generate randomly activated graphs by letting each edge in the graph of Fig. 2a connect with probability 0.5 at each time, and we choose λ to be 4.

Fig. 2b, c, d depict respectively the trajectories of the 5 agents from $k = 1$ to 1500 over the static graph, uniformly jointly connected graphs and randomly activate graphs, where the filled circles are the initial positions of the agents and the black triangle is the geometric median of these circles computed by Weiszfeld's method [1]. As shown in the figures, agents in all cases converge to the geometric median with however slightly different transient performances.

If λ is smaller than the lower bound provided in Theorem 1, consensus may not be achieved among agents. Figure 3 shows the trajectories of agents with $\lambda = 0.8, 2, 1.5$ over a static graph, uniformly jointly connected graphs, randomly activated graphs, respectively. Other settings remain the same except that we increase the times of iterations to 5000. Clearly, agents fail to converge to the geometric median.

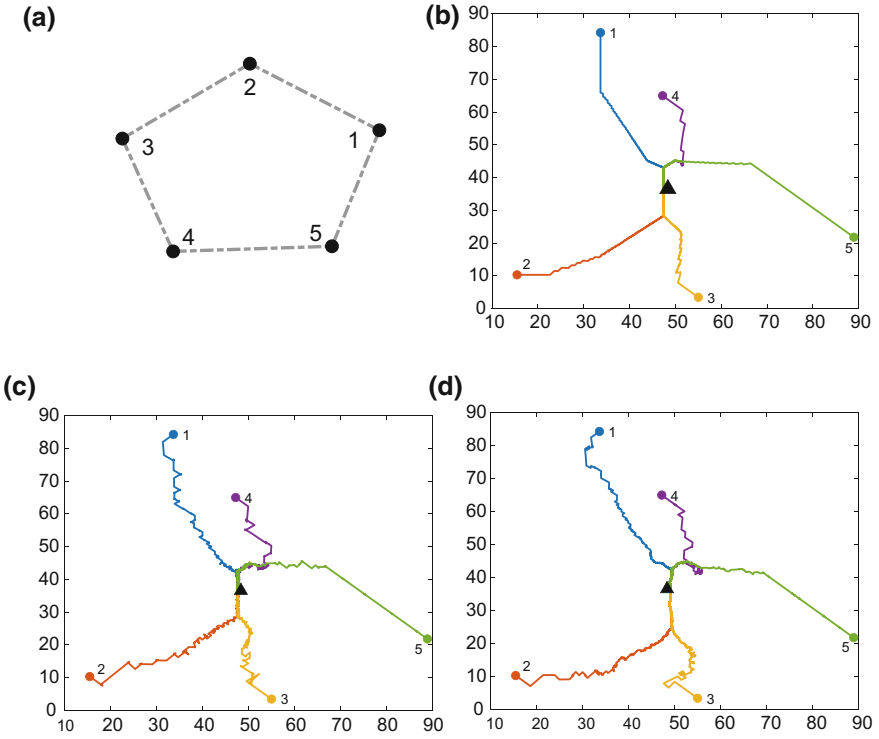

Fig. 2 **a** The topology of the agents. **b** The trajectories of the agents in a static graph, where the filled circles are the initial positions of the agents and the black triangle is the geometric median of these circles. **c** The trajectories of the agents in uniformly jointly connected graphs. **d** The trajectories of the agents in randomly activated graphs

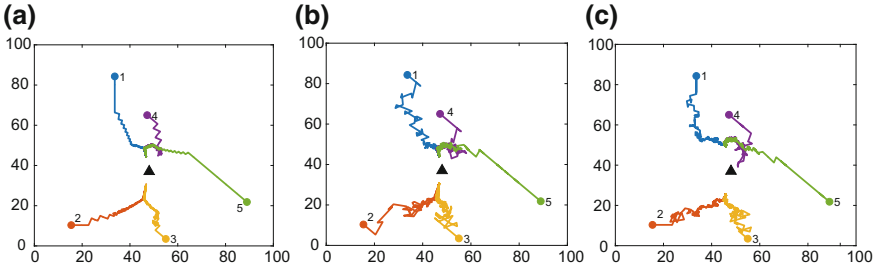

Fig. 3 The trajectories of agents with smaller λ over a static graph, uniformly jointly connected graphs, randomly activated graphs, respectively

7 Conclusions

In this chapter, we have proposed a distributed optimization algorithm to solve the additive cost optimization problem in multi agent networks. Each agent in the algorithm uses only the sign of relative state to each of its neighbor agents. The network was allowed to be static or time-varying. For the former case, we have first provided a penalty method interpretation of our algorithm, and then studied its convergence under diminishing stepsizes as well as a constant stepsize. We have shown that the convergence rate varies from $O(1/\ln(k))$ to $O(1/\sqrt{k})$, depending on the stepsize. For the latter case, we studied the performance of our algorithm over the so-called uniformly jointly connected graphs and randomly activated graphs, the convergence of which is also guaranteed. Finally, we have applied our algorithm to solve a geometric median problem. All the theoretical results have been corroborated via simulations.

Acknowledgements The authors would very much like to thank Professor Tamer Başar for the stimulating discussions on this topic. This work was supported by the National Natural Science Foundation of China under Grant No. 61722308, and National Key Research and Development Program of China under Grant No. 2017YFC0805310.

Appendix: Proof of Theorem 4

We first show that $\widetilde{d}(\rho) < \infty$. Since $\widetilde{f}_\lambda(x)$ is convex, $\widetilde{\mathscr{X}}(\rho)$ is convex and $\mathscr{X}^\star \subseteq \widetilde{\mathscr{X}}(\rho)$ for any $\rho > 0$. One can verify that $\widetilde{\mathscr{X}}(\rho) - \mathscr{X}^\star$ is bounded. If $\widetilde{\mathscr{X}}(\rho) - \mathscr{X}^\star$ is empty, then $\widetilde{d}(\rho) = 0$, otherwise $0 \leq \widetilde{d}(\rho) = \max_{x \in \widetilde{\mathscr{X}}(\rho)} \min_{x^\star \in \mathscr{X}^\star} |x - x^\star| = \max_{x \in \widetilde{\mathscr{X}}(\rho) - \mathscr{X}^\star} \min_{x^\star \in \mathscr{X}^\star} |x - x^\star| < \infty$.

Then, we claim the following.

Claim 1: If $\|\mathbf{x}^k - x^\star \mathbb{1}\| > c_\rho$ for all $x^\star \in \mathscr{X}^\star$, then $\widetilde{f}_\lambda(\mathbf{x}^k) - f^\star > \rho c_a^2/2$.

Recall from (15) that

$$\widetilde{f}_\lambda(\mathbf{x}^k) - f^\star \geq f(\bar{x}^k) - f^\star + (\lambda a_{\min}^{(l)} - \frac{1}{2}cn) v(\mathbf{x}^k), \forall k.$$

This implies that if either $f(\bar{x}^k) - f^\star > \rho c_a^2/2$ or $v(\mathbf{x}^k) > \frac{\rho c_a^2}{2\lambda a_{\min}^{(l)} - cn}$, then $\widetilde{f}_\lambda(\mathbf{x}^k) - f^\star > \rho c_a^2/2$. Let

$$c_\rho := 2\sqrt{n} \max\{\widetilde{d}(\rho), \frac{\rho c_a^2}{2\lambda a_{\min}^{(l)} - cn}\}.$$

Since

$$\begin{aligned} c_\rho &< \|\mathbf{x}^k - x^\star \mathbb{1}\| \leq \|\mathbf{x}^k - \bar{x}^k \mathbb{1}\| + \|\bar{x}^k \mathbb{1} - x^\star \mathbb{1}\| \\ &\leq \sqrt{n} v(\mathbf{x}^k) + \sqrt{n} |\bar{x}^k - x^\star| \end{aligned}$$

we obtain that $v(\mathbf{x}^k) > c_\rho/(2\sqrt{n}) \geq \frac{\rho c_a^2}{2\lambda a_{\min}^{(l)} - cn}$ or $|\bar{x}^k - x^\star| > c_\rho/(2\sqrt{n}) \geq \widetilde{d}(\rho)$. For the former case we have $\widetilde{f}_\lambda(\mathbf{x}^k) - f^\star > \rho c_a^2/2$. For the latter case, $\bar{x}^k \notin \widetilde{\mathscr{X}}(\rho)$, which by the definition of $\widetilde{\mathscr{X}}(\rho)$ implies $\widetilde{f}_\lambda(\mathbf{x}^k) - f^\star > \rho c_a^2/2$.

Claim 2: There is $x_0^\star \in \mathscr{X}^\star$ such that $\liminf_{k\to\infty} \|\mathbf{x}^k - x_0^\star \mathbb{1}\| \leq c_\rho$.

Otherwise, there exists $k_0 > 0$ such that

$$\|\mathbf{x}^k - x^\star \mathbb{1}\| > c_\rho, \forall x^\star \in \mathscr{X}^\star, \forall k > k_0.$$

By Claim 1, there exists some $\varepsilon > 0$ such that $\widetilde{f}_\lambda(\mathbf{x}^k) - f^\star > \rho c_a^2/2 + \varepsilon$ for all $k > k_0$. Together with (18), it yields that

$$\begin{aligned} \|\mathbf{x}^{k+1} - x^\star \mathbb{1}\|^2 &\leq \|\mathbf{x}^k - x^\star \mathbb{1}\|^2 - 2\rho(\widetilde{f}_\lambda(\mathbf{x}^k) - f^\star) + \rho^2 c_a^2 \\ &\leq \|\mathbf{x}^k - x^\star \mathbb{1}\|^2 - 2\rho(\frac{\rho c_a^2}{2} + \varepsilon) + \rho^2 c_a^2 \\ &= \|\mathbf{x}^k - x^\star \mathbb{1}\|^2 - 2\rho\varepsilon. \end{aligned} \tag{38}$$

Summing this relation implies that for all $k > k_0$,

$$\|\mathbf{x}^{k+1} - x^\star \mathbb{1}\|^2 \leq \|\mathbf{x}^{k_0} - x^\star \mathbb{1}\|^2 - 2(k + 1 - k_0)\rho\varepsilon,$$

which clearly cannot hold for a sufficiently large k. Thus, we have verified Claim 2.

Claim 3: There is $x^\star \in \mathscr{X}^\star$ such that $\limsup_{k\to\infty} \|\mathbf{x}^k - x^\star \mathbb{1}\| \leq c_\rho + \rho c_a$.

Otherwise, for *any* $x^\star \in \mathscr{X}^\star$, there must exist a subsequence $\{\mathbf{x}^k\}_{k\in\mathscr{K}}$ (which depends on $x^\star$) such that for all $k \in \mathscr{K}$,

$$\|\mathbf{x}^k - x^\star \mathbb{1}\| > c_\rho + \rho c_a. \tag{39}$$

Notice that the penalty function $h(\mathbf{x})$ can be represented as

$$h(\mathbf{x}) = \sum_{e=1}^{m} a_e |\mathbf{b}_e^\mathsf{T} \mathbf{x}|.$$

where a_e is the weight of edge e. The subdifferential of $h(\mathbf{x})$ is then given by

$$\partial h(\mathbf{x}) = \sum_{e=1}^{m} a_e \mathrm{SGN}(\mathbf{b}_e^\mathsf{T}\mathbf{x})\mathbf{b}_e = B A_e \mathrm{SGN}(B^\mathsf{T}\mathbf{x}) \tag{40}$$

where $A_e = \mathrm{diag}\{a_1, ..., a_m\}$. Then, it follows from (40) that

$$\begin{aligned} \|\mathbf{x}^{k+1} - x^\star \mathbb{1}\| &= \|\mathbf{x}^k - x^\star \mathbb{1} - \rho\lambda B A_e \mathrm{sgn}(B^\mathsf{T}\mathbf{x}^k) - \rho\nabla g(\mathbf{x}^k)\| \\ &\leq \|\mathbf{x}^k - x^\star \mathbb{1}\| + \lambda\rho\|B A_e \mathrm{sgn}(B^\mathsf{T}\mathbf{x}^k)\| + \rho\|\nabla g(\mathbf{x}^k)\| \\ &\leq \|\mathbf{x}^k - x^\star \mathbb{1}\| + \rho\sqrt{n}(\lambda\|A\|_\infty + c) \end{aligned}$$

$$= \|\mathbf{x}^k - x^\star \mathbb{1}\| + \rho c_a, \forall k$$

where the second inequality follows from

$$\begin{aligned}\|BA_e \mathrm{sgn}(B^\mathsf{T}\mathbf{x}^k)\| &\leq \sqrt{n}\|BA_e \mathrm{sgn}(B^\mathsf{T}\mathbf{x}^k)\|_\infty \\ &\leq \sqrt{n}\|BA_e\|_\infty \|\mathrm{sgn}(B^\mathsf{T}\mathbf{x}^k)\|_\infty \\ &\leq \sqrt{n}\max_i \sum_{j=1}^n a_{ij} = \sqrt{n}\|A\|_\infty.\end{aligned}$$

Thus, we obtain that for all $k \in \mathscr{K}$,

$$\|\mathbf{x}^{k-1} - x^\star \mathbb{1}\| \geq \|\mathbf{x}^k - x^\star \mathbb{1}\| - \rho c_a > c_\rho. \tag{41}$$

By Claim 2, there must exist some $k_1 \in \mathscr{K}$ and $k_1 > k_0$ such that

$$\|\mathbf{x}^{k_1-1} - x_0^\star \mathbb{1}\| \leq c_\rho + \rho c_a.$$

Together with (41), it implies that

$$c_\rho < \|\mathbf{x}^{k_1-1} - x_0^\star \mathbb{1}\| \leq c_\rho + \rho c_a. \tag{42}$$

Hence, it follows from Claim 1 that $\widetilde{f}_\lambda(\mathbf{x}^{k_1-1}) - f^\star > \rho c_a^2/2$, which together with (38) and (42) yields that

$$\|\mathbf{x}^{k_1} - x_0^\star \mathbb{1}\| \leq \|\mathbf{x}^{k_1-1} - x_0^\star \mathbb{1}\| \leq c_\rho + \rho c_a. \tag{43}$$

Setting $x^\star = x_0^\star$ in (39), we have $\|\mathbf{x}^{k_1} - x_0^\star \mathbb{1}\| > c_\rho + \rho c_a$. This contradicts (43), and hence verifies Claim 3.

In view of (19), the proof is completed. □

References

1. Beck A, Sabach S (2015) Weiszfelds method: Old and new results. Journal of Optimization Theory and Applications 164(1):1–40
2. Bertsekas DP (2015) Convex Optimization Algorithms. Athena Scientific Belmont
3. Borkar VS (2008) Stochastic approximation: a dynamical systems viewpoint. Baptism's 91 Witnesses
4. Cevher V, Becker S, Schmidt M (2014) Convex optimization for big data: Scalable, randomized, and parallel algorithms for big data analytics. IEEE Signal Processing Magazine 31(5):32–43
5. Chen G, Lewis FL, Xie L (2011) Finite-time distributed consensus via binary control protocols. Automatica 47(9):1962–1968
6. Clarke FH, Ledyaev YS, Stern RJ, Wolenski PR (2008) Nonsmooth Analysis and Control Theory, vol 178. Springer Science & Business Media

7. Cohen MB, Lee YT, Miller G, Pachocki J, Sidford A (2016) Geometric median in nearly linear time. In: Proceedings of the forty-eighth annual ACM symposium on Theory of Computing, ACM, pp 9–21
8. Deo N (1974) Graph Theory with Applications to Engineering and Computer Science. Courier Dover Publications
9. Franceschelli M, Giua A, Pisano A (2017) Finite-time consensus on the median value with robustness properties. IEEE Transactions on Automatic Control 62(4):1652–1667
10. Kan Z, Shea JM, Dixon WE (2016) Leader–follower containment control over directed random graphs. Automatica 66:56–62
11. Li T, Fu M, Xie L, Zhang J (2011) Distributed consensus with limited communication data rate. IEEE Transactions on Automatic Control 56(2):279–292
12. Lin P, Ren W, Farrell JA (2017) Distributed continuous-time optimization: nonuniform gradient gains, finite-time convergence, and convex constraint set. IEEE Transactions on Automatic Control 62(5):2239–2253
13. Magnússon S, Enyioha C, Li N, Fischione C, Tarokh V (2017) Convergence of limited communications gradient methods. IEEE Transactions on Automatic Control 63(5):1356–1371
14. Mokhtari A, Ling Q, Ribeiro A (2017) Network Newton distributed optimization methods. IEEE Transactions on Signal Processing 65(1):146–161
15. Nedić A, Olshevsky A (2015) Distributed optimization over time-varying directed graphs. IEEE Transactions on Automatic Control 60(3):601–615
16. Nedic A, Ozdaglar A (2009) Distributed subgradient methods for multi-agent optimization. IEEE Transactions on Automatic Control 54(1):48–61
17. Nedić A, Olshevsky A, Rabbat MG (2017) Network topology and communication-computation tradeoffs in decentralized optimization. In: Proceedings of the IEEE, vol 106, no 5, pp 953–976, May 2018
18. Olfati-Saber R, Murray RM (2004) Consensus problems in networks of agents with switching topology and time-delays. IEEE Transactions on Automatic Control 49(9):1520–1533
19. Pu Y, Zeilinger MN, Jones CN (2017) Quantization design for distributed optimization. IEEE Transactions on Automatic Control 62(5):2107–2120
20. Shi W, Ling Q, Wu G, Yin W (2015) Extra: An exact first-order algorithm for decentralized consensus optimization. SIAM Journal on Optimization 25(2):944–966
21. Wang H, Li C (2017) Distributed quantile regression over sensor networks. IEEE Transactions on Signal and Information Processing over Networks pp 1–1, 10.1109/TSIPN.2017.2699923
22. Xie P, You K, Tempo R, Song S, Wu C (2018) Distributed convex optimization with inequality constraints over time-varying unbalanced digraphs. IEEE Transactions on Automatic Control PP(99):1–1, 10.1109/TAC.2018.2816104
23. Yi P, Hong Y (2014) Quantized subgradient algorithm and data-rate analysis for distributed optimization. IEEE Transactions on Control of Network Systems 1(4):380–392
24. You K, Xie L (2011) Network topology and communication data rate for consensusability of discrete-time multi-agent systems. IEEE Transactions on Automatic Control 56(10):2262–2275
25. You K, Tempo R, Xie P (2018) Distributed algorithms for robust convex optimization via the scenario approach. IEEE Transactions on Automatic Control
26. Zhang J, You K (2018) Distributed optimization with binary relative information over deterministically time-varying graphs. To appear in the 57th IEEE Conference on Decision and Control, Miami Beach, FL, USA
27. Zhang J, You K, Başar T (2017) Distributed discrete-time optimization by exchanging one bit of information. In: 2018 annual American Control Conference (ACC), IEEE, pp 2065–2070
28. Zhang J, You K, Başar T (2018) Distributed discrete-time optimization in multi-agent networks using only sign of relative state. Accepted by IEEE Transactions on Automatic Control