To appear in *Optimization*
Vol. 00, No. 00, Month 20XX, 1–27

Research Article

Almost Sure Convergence of Random Projected Proximal and Subgradient Algorithms for Distributed Nonsmooth Convex Optimization

Hideaki Iiduka[a*]

[a] *Department of Computer Science, Meiji University, 1-1-1 Higashimita, Tama-ku, Kawasaki-shi, Kanagawa 214-8571 Japan*
iiduka@cs.meiji.ac.jp

(*Received 00 Month 20XX; accepted 00 Month 20XX*)

Two distributed algorithms are described that enable all users connected over a network to cooperatively solve the problem of minimizing the sum of all users' objective functions over the intersection of all users' constraint sets, where each user has its own private nonsmooth convex objective function and closed convex constraint set, which is the intersection of a number of simple, closed convex sets. One algorithm enables each user to adjust its estimate by using the proximity operator of its objective function and the metric projection onto one constraint set randomly selected from a number of simple, closed convex sets. The other determines each user's estimate by using the subdifferential of its objective function instead of the proximity operator. Investigation of the two algorithms' convergence properties for a diminishing step-size rule revealed that, under certain assumptions, the sequences of all users generated by each of the two algorithms converge almost surely to the same solution. It also showed that the rate of convergence depends on the step size and that a smaller step size results in quicker convergence. The results of numerical evaluation using a nonsmooth convex optimization problem support the convergence analysis and demonstrate the effectiveness of the two algorithms.

Keywords: almost sure convergence; distributed nonsmooth convex optimization; metric projection; proximity operator; random projection algorithm; subgradient

AMS Subject Classification: 90C15; 90C25; 90C30

1. Introduction

Future network models have attracted a great deal of attention. The concept of the network model considered here differs completely from that of a conventional client-server network model. While a conventional client-server network model explicitly distinguishes hosts providing services (servers) from hosts receiving services (clients), the network model considered here does not assign fixed roles to hosts. Hosts composing the network, referred to here as users, can be both servers and clients. Hence, the network can function as an *autonomous*, *distributed*, and *cooperative* system. Although there are several forms of networks in which some operations are intentionally centralized (e.g., hybrid peer-to-peer networks), here the focus is on networks that do not have centralized operations. Therefore, distributed mechanisms need to be used that can work in cooperation with each user and neighboring users to control the network.

* This work was supported by the Japan Society for the Promotion of Science through a Grant-in-Aid for Scientific Research (C) (15K04763).

Distributed optimization (see, e.g., [2], [4], [9], [11, Part II], [12, PART II], [27], [28], [33] and references therein) plays a crucial role in making future networks [10, 14, 23, 30], such as wireless, sensor, and peer-to-peer networks, stable and highly reliable. One way [23, Chapter 5], [25], [33, Chapter 2] to achieve this optimization is to model the utility and strategy of each user respectively as a concave utility function and convex constraint set and solve the problem of maximizing the overall utility of all users over the intersection of all users' constraint sets. This paper focuses on a constrained convex minimization problem in which each user has its own *nonsmooth convex* objective function (i.e., a minus nonsmooth concave utility function) and closed convex constraint set that is the intersection of a number of simple, closed convex sets (e.g., the intersection of affine subspaces, half-spaces, and hyperslabs). The constrained nonsmooth convex optimization problem covers the important situations in which each user's objective function is differentiable with a non-Lipschitz continuous gradient or not differentiable (e.g., the L^1-norm) and includes, for instance, the problem of minimizing the total variation of a signal over a convex set, Tykhonov-like problems with L^1-norms [13, I. Introduction], the classifier ensemble problem with sparsity and diversity learning [39, Subsection 2.2.3], [40, Subsection 3.2.4], which is expressed as L^1-norm minimization, and the minimal antenna-subset selection problem [38, Subsection 17.4]. The main objective of the present paper is to show that the constrained nonsmooth convex optimization problem for many real-world applications can be solved by using distributed optimization techniques.

Two distributed optimization algorithms are presented for solving the constrained convex minimization problem described above. At each iteration of the first algorithm, each user calculates the weighted average of its estimate and the estimates received from its neighboring users and then updates its estimate by using the weighted average, the *proximity operator* of its own private nonsmooth convex function, and the metric projection onto one constraint set randomly selected from a number of simple, closed convex sets. The second algorithm is obtained by replacing the proximity operator in the first algorithm with the *subdifferential* of each user's nonsmooth convex function.

The two algorithms are performed on the basis of a framework [24, (2a)] for local user communications and random observations [24, (2b)], [26, (2)] of the local constraints, which ensures that each user can observe one simple, closed convex set onto which the metric projection can be efficiently calculated. Accordingly, the two algorithms can be applied to two complicated cases. In Case 1, each user does not know the full form of its private constraint set in advance and can observe only one simple, closed convex set at each instance. In Case 2, each user knows the full form of its private constraint set in advance, and the constraint set is the intersection of a huge number of simple, closed convex sets, which means that a metric projection onto the constraint set cannot be calculated easily. See [24, Section I] and [26, Section 1] and references therein for details on applications of the two cases, including collaborative filtering for recommender systems and text classification problems.

Proximal point methods and subgradient methods with randomized order [4, Section 4], [27, Section 3] are useful for constrained nonsmooth convex optimization. These methods are implemented under the condition that each user uses a randomly chosen component function at each iteration while the proposed algorithms are performed under the condition that each user uses a randomly chosen closed convex set onto which the metric projection can be efficiently calculated (Subsections 3.2 and 4.2 explicitly compare the methods in [4, Section 4] and [27, Section 3] with the proposed algorithms). Related random projection algorithms have been proposed for convex optimization over the intersection of a number of closed convex sets. The algorithm most relevant to the work reported here is the first distributed random projected gradient algorithm [24] that was proposed for solving a constrained smooth convex minimization problem when each user's objective function is convex with a Lipschitz continuous gradient. The centralized random projected gradient and subgradient algorithms [26] were proposed for minimizing a single objective function over the intersection of an arbitrary family of convex inequalities. The incremental constraint projection-proximal algorithm [36] uses both random subgradient updates and random constraint updates. While there have been no reports on distributed random projection algorithms for nonsmooth convex optimization, thanks to the useful ideas in [24, 36], a distributed random projected proximal algorithm (Algorithm 3.1) can be devised. Moreover, on the basis of [24, 26], a distributed random projected subgradient algorithm (Algorithm 4.1) can be devised that is a generalization of the first distributed random projected gradient algorithm [24, (2a), (2b)]. Furthermore, there has been much work [35] on random projection of vectors in a higher dimensional space to a randomly chosen lower dimensional subspace, and convex programming problems [29] and affine variational inequalities [31] have been analyzed using random projections defined in the same way [35]. Since, with the proposed algorithms, each user is assumed to use a randomly chosen projection at each iteration that projects a vector in the whole space to a closed convex subset of the whole space, the definition of random projection in [29, 31, 35] clearly differs from the one used here.

This paper makes two contributions that build on previously reported results. First, it presents two novel distributed random projection algorithms for constrained nonsmooth convex optimization that are based on each user's local communications. This means that they can be implemented independently of the network topology and that each user can calculate the weighted average of its estimate and the neighboring users' estimates. The algorithms proceed by performing a proximal or subgradient step for each user's objective function at the weighted average and projecting onto one simple, closed convex set that is randomly selected from each user's local constraint sets. Since the metric projection is a special case of nonexpansive mapping, the algorithms are related to previous fixed point optimization algorithms [15, 16, 18, 19, 21] for convex optimization over fixed point sets of nonexpansive mappings. Furthermore, it presents a gradient method [18] that accelerates previously reported optimization algorithms for minimizing one smooth convex objective function over a fixed point set of a nonexpansive mapping. These algorithms [16, 21] are synchronous decentralized algorithms that can be applied to smooth convex optimization over fixed point sets of nonexpansive mappings. Since these algorithms [16, 18, 21] work for only smooth convex optimization, they cannot be applied to the constrained nonsmooth convex optimization problem considered here. Parallel and incremental subgradient algorithms [15] have been proposed to solve the problem of minimizing the sum of nonsmooth, convex objective functions over fixed point sets, and incremental proximal point algorithms [19] have been proposed for solving the problem. These algorithms [15, 19] work for nonsmooth convex optimization over the intersection of convex constraint sets that are not always simple. However, since they work only when each user makes the best use of its own private information, they cannot be applied to the case in which each user does not know the explicit form of its constraint set in advance. Moreover, since these algorithms [15, 19] can be applied only to deterministic optimization, they cannot be applied to the case in which each user uses one constraint set selected randomly at each iteration. In contrast, the proposed algorithms work even when each user randomly sets one projection selected from many projections (see Case 1 description above). Therefore, the proposed algorithms have wider application than previous fixed point optimization algorithms [15, 16, 18, 19, 21].

The second contribution is an analysis of the proposed proximal and subgradient algorithms. In contrast to the convergence analysis of the first distributed random projected gradient algorithm [24], smooth convex analysis, which has tractable properties due to the use of Lipschitz continuous gradients, cannot be applied to the convergence analyses of the two proposed algorithms, which optimize nonsmooth convex functions. However, convergence analyses of the two algorithms can be performed by using useful properties [1, Propositions 12.16, 12.27, and 16.14] (Proposition 2.1) of the proximity operators and the subgradients of nonsmooth convex objective functions. Thanks to the supermartingale convergence theorem [5, Proposition 8.2.10] (Proposition 2.2) and the portmanteau lemma [6, Theorem 16], [34, Lemma 2.2] (Proposition 2.3), it is guaranteed that, under certain assumptions, the sequences of all users generated by each of the two algorithms converge almost surely to the same solution to the constrained nonsmooth convex optimization problem considered in this paper (Theorems 3.1 and 4.1). Moreover, the rates of convergence of the two algorithms (Proposition 3.1, (11), Proposition 4.1, and (22)) are provided to illustrate their algorithm efficiencies. The convergence rate analysis leads to selection of a step size such that the two algorithms converge quickly. Numerical results for the two algorithms are provided to support the convergence and convergence rate analyses.

This paper is organized as follows. Section 2 gives the mathematical preliminaries and states the main problem. Section 3 presents the proposed random projected proximal algorithm for solving the main problem and describes its convergence properties for a diminishing step size. Section 4 presents the proposed random projected subgradient algorithm for solving the main problem and describes its convergence properties for a diminishing step size. Section 5 presents a numerical evaluation using a nonsmooth convex optimization problem and compares the behaviors of the two algorithms. Section 6 concludes the paper with a brief summary and mentions future directions for improving the proposed algorithms.

2. Preliminaries

2.1. *Definitions and propositions*

Let $\mathbb{R}^d$ be a d-dimensional Euclidean space with inner product $\langle \cdot, \cdot \rangle$ and its induced norm $\|\cdot\|$, and let $\mathbb{R}^d_+ := \{(x_1, x_2, \ldots, x_d) \in \mathbb{R}^d \colon x_i \geq 0 \ (i = 1, 2, \ldots, d)\}$. Let $[W]_{ij}$ and $W^\top$ denote the (i,j)th entry and the transpose of a matrix W. Let $\Pr\{X\}$ and $\mathrm{E}[X]$ denote the probability and the expectation of a random variable X. The probability space considered here is denoted by $(\Omega, \mathcal{F}, \Pr)$.

The metric projection onto a nonempty, closed convex set $C \subset \mathbb{R}^d$ is denoted by P_C, and it is defined for all $x \in \mathbb{R}^d$ by $P_C(x) \in C$ and $\|x - P_C(x)\| = \mathrm{d}(x, C) := \inf\{\|x - y\| \colon y \in C\}$. Mapping P_C satisfies the firm nonexpansivity condition [1, Proposition 4.8]; i.e., $\|P_C(x) - P_C(y)\|^2 + \|(x - P_C(x)) - (y - P_C(y))\|^2 \leq \|x - y\|^2$ $(x, y \in \mathbb{R}^d)$. This means that $\|P_C(x) - P_C(y)\| \leq \|x - y\|$ $(x, y \in \mathbb{R}^d)$; i.e., P_C is nonexpansive. The subdifferential [1, Definition 16.1] of $f \colon \mathbb{R}^d \to \mathbb{R}$ is the set-valued operator defined for all $x \in \mathbb{R}^d$ by $\partial f(x) := \{u \in \mathbb{R}^d \colon f(y) \geq f(x) + \langle y - x, u \rangle \ (y \in \mathbb{R}^d)\}$. The proximity operator [1, Definition 12.23] of a convex function $f \colon \mathbb{R}^d \to \mathbb{R}$, denoted by prox_f, maps every $x \in \mathbb{R}^d$ to the unique minimizer of $f(\cdot) + (1/2)\|x - \cdot\|^2$;

i.e., $\mathrm{prox}_f(x) \in \mathrm{Argmin}_{y\in\mathbb{R}^d}[f(y) + (1/2)\|x - y\|^2]$ ($x \in \mathbb{R}^d$). The uniqueness and existence of $\mathrm{prox}_f(x)$ are guaranteed for all $x \in \mathbb{R}^d$ [1, Definition 12.23].

Proposition 2.1 Let $f : \mathbb{R}^d \to \mathbb{R}$ be convex. Then the following hold:

(i) $\partial f(x)$ is nonempty for all $x \in \mathbb{R}^d$.
(ii) Let $x, p \in \mathbb{R}^d$. $p = \mathrm{prox}_f(x)$ if and only if $x - p \in \partial f(p)$ (i.e., $\langle y - p, x - p\rangle + f(p) \leq f(y)$ for all $y \in \mathbb{R}^d$).
(iii) prox_f is firmly nonexpansive.
(iv) Let $L > 0$. Then f is Lipschitz continuous with a Lipschitz constant L if and only if $\|u\| \leq L$ for all $x \in \mathbb{R}^d$ and for all $u \in \partial f(x)$.

Proof: (i)–(iii) Propositions 16.14(ii) and 12.26 in [1] lead to (i) and (ii) while Proposition 12.27 in [1] implies (iii).

(iv) Theorem 6.2.2, Corollary 6.1.2, and Exercise 6.1.9(c) in [7] lead to (iv). □

The following propositions are needed to prove the main theorems.

Proposition 2.2 [The supermartingale convergence theorem [5, Proposition 8.2.10]] Let $(Y_k)_{k\geq0}$, $(Z_k)_{k\geq0}$, and $(W_k)_{k\geq0}$ be sequences of nonnegative random variables, and let $\mathcal{F}_k$ ($k \geq 0$) denote the collection $Y_0, Y_1, \ldots, Y_k$, $Z_0, Z_1, \ldots, Z_k$, and $W_0, W_1, \ldots, W_k$. Suppose that $\sum_{k=0}^{\infty} W_k < \infty$ almost surely and that almost surely, for all $k \geq 0$, $\mathrm{E}[Y_{k+1}|\mathcal{F}_k] \leq Y_k - Z_k + W_k$. Then $\sum_{k=0}^{\infty} Z_k < \infty$ almost surely and $(Y_k)_{k\geq0}$ converges almost surely to a nonnegative random variable Y.

Proposition 2.3 [The portmanteau lemma [6, Theorem 16], [34, Lemma 2.2]] Let $(Y_k)_{k\geq0}$ be a sequence of random variables that converges in law to a random variable Y. Then $\limsup_{k\to\infty} \Pr\{Y_k \in F\} \leq \Pr\{Y \in F\}$ for every closed set F.

A directed graph $G := (V, E)$ is a finite nonempty set V of nodes (users) and a collection E of ordered pairs of distinct nodes from V [3, p. 394]. A directed graph is said to be strongly connected if, for each pair of nodes i and l, there exists a directed path from i to l [3, p. 394].

2.2. *Main problem and assumptions*

Let us consider a constrained nonsmooth convex optimization problem that is distributed over a network of m users, indexed by $V := \{1, 2, \ldots, m\}$. User i ($i \in V$) has its own private function $f_i : \mathbb{R}^d \to \mathbb{R}$ and constraint set $X_i \subset \mathbb{R}^d$. On the basis of [24, Section II], let us define the constraint set $X := \bigcap_{i=1}^m X_i$ for the whole network. Suppose that X is the intersection of n closed convex sets. Let $I := \{1, 2, \ldots, n\}$, and let I_i ($i \in V$) be the partition of I such that $I = \bigcup_{i=1}^m I_i$ and $I_i \cap I_l = \emptyset$ for all $i, l \in V$ with $i \neq l$. Then X_i can be defined by the intersection of closed convex sets X_i^j ($j \in I_i$); i.e., $X_i := \bigcap_{j\in I_i} X_i^j$ ($i \in V$) and $X := \bigcap_{i\in V} X_i = \bigcap_{i=1}^m \bigcap_{j\in I_i} X_i^j$. The following is assumed hereafter.

Assumption 2.1 Suppose that

(A1) $X_i^j \subset \mathbb{R}^d$ ($i \in V, j \in I_i$) is a closed convex set onto which the metric projection $P_{X_i^j}$ can be efficiently computed, and $X := \bigcap_{i\in V} X_i \neq \emptyset$;
(A2) $f_i : \mathbb{R}^d \to \mathbb{R}$ ($i \in V$) is convex;
(A3) For all $i \in V$, there exists $M_i \in \mathbb{R}$ such that $\sup\{\|g_i\| : x \in X_i, g_i \in \partial f_i(x)\} \leq M_i$.

Assumption (A3) is satisfied if f_i ($i \in V$) is polyhedral on X_i or X_i ($i \in V$) is bounded [5, p. 471]. Proposition 2.1(iv) guarantees that, if f_i $(i \in V)$ is Lipschitz continuous on X_i, Assumption (A3) holds.

The following is the main problem discussed here.

Problem 2.1 Under Assumption 2.1,

$$\text{minimize } f(x) := \sum_{i \in V} f_i(x) \text{ subject to } x \in X := \bigcap_{i \in V} X_i,$$

where one assumes that Problem 2.1 has a solution.

The solution set of Problem 2.1 is denoted by $X^\star := \{x^\star \in X : f(x^\star) = f^\star := \inf_{x \in X} f(x)\}$. The condition $X^\star \neq \emptyset$ holds, for example, when one of X_i^js is bounded [1, Corollary 8.31, Proposition 11.14].

The main objective here is to present distributed optimization algorithms that enable each user to solve Problem 2.1 without using other users' private information. This goal is addressed by assuming that each user and its neighboring users form a network in which each user can transmit its estimate to its neighboring users. The network topology at time k is expressed as a directed graph $G(k) := (V, E(k))$, where $E(k) \subset V \times V$, and $(i, j) \in E(k)$ stands for a link such that user i receives information from user j at time k. Let $N_i(k) \subset V$ be the set of users that send information to user i; i.e., $N_i(k) := \{j \in V : (i, j) \in E(k)\}$ and $i \in N_i(k)$ $(i \in V, k \geq 0)$. To consider Problem 2.1, the following assumptions are needed [24, Assumptions 4 and 5], which leads to Lemma 3.3 (see [32, Theorem 4.2]) used to prove the main theorems.

Assumption 2.2 There exists $Q \geq 1$ such that the graph $(V, \bigcup_{l=0}^{Q-1} E(k+l))$ is strongly connected for all $k \geq 0$.

Assumption 2.3 For $k \geq 0$, user i $(i \in V)$ has the weighted parameters $w_{ij}(k)$ $(j \in V)$ satisfying the following:

(i) $w_{ij}(k) := [W(k)]_{ij} \geq 0$ for all $j \in V$, and $w_{ij}(k) = 0$ when $j \notin N_i(k)$;
(ii) There exists $w \in (0, 1)$ such that $w_{ij}(k) \geq w$ for all $j \in N_i(k)$.
(iii) $\sum_{j \in V} [W(k)]_{ij} = 1$ for all $i \in V$ and $\sum_{i \in V} [W(k)]_{ij} = 1$ for all $j \in V$.

Moreover, user i $(i \in V)$ has the step size $(\alpha_k)_{k \geq 0} \subset (0, \infty)$ satisfying (C1) $\sum_{k=0}^{\infty} \alpha_k = \infty$ and (C2) $\sum_{k=0}^{\infty} \alpha_k^2 < \infty$.

The matrix $W(k)$ $(k \geq 0)$ satisfying Assumption 2.3(i)–(iii) is said to be *doubly stochastic* [24, Assumption 5]. Theorem 1 in [37] indicates that the spectral radius ρ of the doubly stochastic matrix $W(k)$ $(k \geq 0)$ is less than 1 if and only if, for all $k \geq 0$,

$$\lim_{s \to \infty} W(k)^s = \frac{ee^\top}{m}, \tag{1}$$

where $e \in \mathbb{R}^d$ stands for the column vector in which all entries are equal to 1. Moreover, Theorem 3.2.1 in [8] and the remark in [37, Theorem 1] show that (1) means the network is strongly connected. Accordingly, Assumption 2.2 holds if $\rho < 1$ (e.g., $\rho < 1$ holds when one is a simple eigenvalue of $W(k)$ and all other eigenvalues are strictly less than one in magnitude [37, p. 67]). See [17, Examples 2.6 and 2.7] for examples of $W(k)$ satisfying Assumption 2.3(i)–(iii).

Assumptions (A1) and (A2) imply that, for all $k \geq 0$, user i $(i \in V)$ can determine its estimate by using a subdifferential or proximity operator of f_i and the metric projection onto a certain constraint set selected from its own constraint sets X_i^j ($j \in I_i$). Here it is assumed that user i ($i \in V$) forms the metric projection on the basis of random observations of the local constraints; i.e., user i observes a local constraint set at time k, $X_i^{\Omega_i(k)}$, where $\Omega_i(k) \in I_i$ is a random variable, and thus uses $P_{X_i^{\Omega_i(k)}}$. Convergence analyses are performed by assuming the following [24, Assumptions 2 and 3]:

Assumption 2.4 The random sequences $(\Omega_i(k))_{k\geq 0}$ ($i \in V$) are independent and identically distributed and independent of the initial points $x_i(0)$ ($i \in V$) in the algorithms presented here. Moreover, $\Pr\{\Omega_i(k) = j\} > 0$ holds for $i \in V$ and for $j \in I_i$.

Assumption 2.5 There exists $c > 0$ such that, for all $i \in V$, for all $x \in \mathbb{R}^d$, and for all $k \geq 0$, $\mathrm{d}(x, X)^2 \leq c\mathrm{E}[\mathrm{d}(x, X_i^{\Omega_i(k)})^2]$.

The sequences $(\Omega_i(k))_{k\geq 0}$ ($i \in V$) satisfy Assumption 2.4 when the sample constraints are generated through state transitions of a Markov chain [36, Subsection 4.4]. Assumption 2.4 is satisfied when the constraints are sampled in a cyclic manner by random shuffling or in accordance with a deterministic cyclic order [36, Subsection 4.3]. Other examples of $(\Omega_i(k))_{k\geq 0}$ satisfying Assumption 2.4 are described in Subsections 4.1 and 4.2 in [36]. Assumption 2.5 holds when X_i^j ($i \in V, j \in I_i$) is a linear inequality or an equality constraint, or when X has an interior point (see [24, p. 223] and references therein). Even if X_i^j ($i \in V, j \in I_i$) is a restricted constraint such as a linear inequality or equality constraint, $X_i := \bigcap_{j \in I_i} X_i^j$ ($i \in V$) is complicated in the sense that metric projection onto X_i is not necessarily easy. Accordingly, the metric projection onto the whole constraint set $X := \bigcap_{i\in V} X_i$ cannot be efficiently computed, and hence, $X := \bigcap_{i \in V} \bigcap_{j\in I_i} X_i^j$ has a complicated form even when X_i^j is simple enough to have a closed form expression of the metric projection onto X_i^j. The proposed algorithms (Algorithms 3.1 and 4.1) using the metric projection onto X_i^j can be applied to Problem 2.1 with such a complicated constraint X. In contrast, the previously reported algorithms [4, Section 4], [27, Section 3] need to use metric projection onto the whole constraint set X (see Subsections 3.2 and 4.2 for detailed comparisons of the proposed algorithms with the previous ones [4, Section 4], [27, Section 3]).

Let $x_i(k) \in \mathbb{R}^d$ be the estimate of user i at time k (see (4) and (16) in Algorithms 3.1 and 4.1 for details of the definition of $(x_i(k))_{k\geq 0}$ ($i \in V$)). The proposed algorithms are analyzed by using the expectation taken with respect to the past history of the algorithms defined as follows [24, p. 224]. Let $\mathcal{F}_k$ be the σ-algebra generated by the entire history of the algorithms up to time $(k-1)$ inclusively; i.e., $\mathcal{F}_0 := \sigma(x_i(0)\ (i \in V))$, and for all $k \geq 1$,

$$\mathcal{F}_k := \sigma\left(x_i\left(0\right)\ \left(i \in V\right),\ \Omega_i\left(l\right)\ \left(l \in \left[0, k-1\right],\ i \in V\right)\right). \tag{2}$$

3. Distributed random projected proximal algorithm

This section presents the proposed proximal algorithm using random projections for solving Problem 2.1 under Assumptions 2.1–2.3.

Algorithm 3.1

Step 0. User i ($i \in V$) sets $x_i(0)$ arbitrarily.
Step 1. User i ($i \in V$) receives $x_j(k)$ from its neighboring users $j \in N_i(k)$ and computes the weighted average

$$v_i(k) := \sum_{j \in N_i(k)} w_{ij}(k) x_j(k). \tag{3}$$

User i updates its estimate $x_i(k+1)$ by using

$$x_i(k+1) := P_{X_i^{\Omega_i(k)}}\left(\mathrm{prox}_{\alpha_k f_i}(v_i(k))\right). \tag{4}$$

The algorithm sets $k := k+1$ and returns to Step 1.

Definition (2) implies that, given $\mathcal{F}_k$ ($k \geq 0$), the collection $x_i(0), x_i(1), \ldots, x_i(k)$ and $v_i(0), v_i(1), \ldots, v_i(k)$ generated by Algorithm 3.1 is fully determined. Algorithm 3.1 enables user i ($i \in V$) to determine $x_i(k+1)$ by using its proximity operator for the weighted average $v_i(k)$ of the received $x_j(k)$ ($j \in N_i(k)$) and the metric projection onto a constraint set $X_i^{\Omega_i(k)}$ randomly selected from X_i^j.

The convergence analysis of Algorithm 3.1 is as follows.

Theorem 3.1 Under Assumptions 2.1–2.3, the sequence $(x_i(k))_{k \geq 0}$ ($i \in V$) generated by Algorithm 3.1 converges almost surely to a random point $x^\star \in X^\star$.

3.1. *Proof of Theorem 3.1*

The following lemma can be proven by using Proposition 2.1 and the firm nonexpansivity condition of $P_{X_i^{\Omega_i(k)}}$ and by referring to the proof of [26, Lemma 2]. Hence, the proof of Lemma 3.1 can be omitted.

Lemma 3.1 Suppose that Assumption 2.1 holds. Then for all $x \in X$, for all $i \in V$, for all $z \in X_i$, for all $k \geq 0$, and for all $\tau, \eta, \mu > 0$,

$$\begin{aligned}\|x_i(k+1) - x\|^2 \leq \|v_i(k) - x\|^2 &- 2\alpha_k (f_i(z) - f_i(x)) + N(\tau, \eta)\alpha_k^2 \\ &+ \tau \|v_i(k) - z\|^2 + (\eta + \mu - 2)\left\|v_i(k) - \mathrm{prox}_{\alpha_k f_i}(v_i(k))\right\|^2 \\ &- \left(1 - \frac{1}{\mu}\right)\|x_i(k+1) - v_i(k)\|^2,\end{aligned}$$

where $N(\tau, \eta) := (1/\tau + 1/\eta) \max_{i \in V} M_i^2 < \infty$ and M_i ($i \in V$) is as in Assumption (A3).

The following lemma is shown.

Lemma 3.2 Suppose that Assumptions 2.1, 2.3, and 2.5 hold. Then $\sum_{k=0}^\infty \|v_i(k) - \mathrm{prox}_{\alpha_k f_i}(v_i(k))\|^2 < \infty$, and $\sum_{k=0}^\infty \mathrm{d}(v_i(k), X)^2 < \infty$ almost surely for all $i \in V$.

Proof: The definition of d means that $\mathrm{d}(v_i(k), X) = \|v_i(k) - P_X(v_i(k))\|$ and $\mathrm{d}(x_i(k+1), X) \leq \|x_i(k+1) - P_X(v_i(k))\|$ ($i \in V, k \geq 0$). The condition $x_i(k+1) \in X_i^{\Omega_i(k)}$ ($i \in V, k \geq 0$) means that $\mathrm{d}(v_i(k), X_i^{\Omega_i(k)}) \leq \|v_i(k) - x_i(k+1)\|$ ($i \in V, k \geq 0$). Then Lemma 3.1 with $x = z := P_X(v_i(k))$ ($i \in V, k \geq 0$) guarantees that, for all $i \in V$, for all $k \geq 0$, for all $\tau, \eta > 0$, and for all $\mu > 1$,

$$\begin{aligned}\mathrm{d}\left(x_i\left(k+1\right), X\right)^2 \leq\ & \mathrm{d}\left(v_i\left(k\right), X\right)^2 + \tau \mathrm{d}\left(v_i\left(k\right), X\right)^2 \\ & + \left(\eta + \mu - 2\right) \left\| v_i\left(k\right) - \mathrm{prox}_{\alpha_k f_i}\left(v_i\left(k\right)\right)\right\|^2 + N\left(\tau, \eta\right) \alpha_k^2 \\ & - \left(1 - \frac{1}{\mu}\right) \mathrm{d}\left(v_i\left(k\right), X_i^{\Omega_i(k)}\right)^2.\end{aligned}$$

By taking the expectation in this inequality conditioned on $\mathcal{F}_k$ defined in (2), Assumption 2.5 leads to the finding that, almost surely, for all $i \in V$, for all $k \geq 0$, for all $\tau, \eta > 0$, and for all $\mu > 1$,

$$\begin{aligned}\mathrm{E}\left[\mathrm{d}\left(x_i\left(k+1\right), X\right)^2 \middle| \mathcal{F}_k\right] \leq\ & \mathrm{d}\left(v_i\left(k\right), X\right)^2 + \left(\eta + \mu - 2\right) \left\| v_i\left(k\right) - \mathrm{prox}_{\alpha_k f_i}\left(v_i\left(k\right)\right)\right\|^2 \\ & + \left\{\tau - \frac{1}{c}\left(1 - \frac{1}{\mu}\right)\right\} \mathrm{d}\left(v_i\left(k\right), X\right)^2 + N\left(\tau, \eta\right) \alpha_k^2.\end{aligned}$$

Let us take $\tau := 1/(6c)$, $\eta := 1/3$, and $\mu := 3/2$. From $\eta + \mu - 2 = -1/6$, $\tau - (1/c)(1 - 1/\mu) = -1/(6c)$, and the convexity of $\mathrm{d}(\cdot, X)^2$, almost surely for all $i \in V$ and for all $k \geq 0$,

$$\begin{aligned}\mathrm{E}\left[\mathrm{d}\left(x_i\left(k+1\right), X\right)^2 \middle| \mathcal{F}_k\right] \leq\ & \sum_{j=1}^{m} \left[W(k)\right]_{ij} \mathrm{d}\left(x_j\left(k\right), X\right)^2 - \frac{1}{6} \left\| v_i\left(k\right) - \mathrm{prox}_{\alpha_k f_i}\left(v_i\left(k\right)\right)\right\|^2 \\ & - \frac{1}{6c} \mathrm{d}\left(v_i\left(k\right), X\right)^2 + N\left(\frac{1}{6c}, \frac{1}{3}\right) \alpha_k^2,\end{aligned}$$

where $N(1/(6c), 1/3) < \infty$ is guaranteed from Assumption (A3). Hence, Assumption 2.3 ensures that, almost surely, for all $k \geq 0$,

$$\begin{aligned}\mathrm{E}\left[\sum_{i=1}^{m} \mathrm{d}\left(x_i\left(k+1\right), X\right)^2 \middle| \mathcal{F}_k\right] \leq\ & \sum_{j=1}^{m} \mathrm{d}\left(x_j\left(k\right), X\right)^2 - \frac{1}{6} \sum_{i=1}^{m} \left\| v_i\left(k\right) - \mathrm{prox}_{\alpha_k f_i}\left(v_i\left(k\right)\right)\right\|^2 \\ & - \frac{1}{6c} \sum_{i=1}^{m} \mathrm{d}\left(v_i\left(k\right), X\right)^2 + mN\left(\frac{1}{6c}, \frac{1}{3}\right) \alpha_k^2. \qquad (5)\end{aligned}$$

Proposition 2.2 and (C2) lead to $\sum_{k=0}^{\infty} \sum_{i=1}^{m} \|v_i(k) - \mathrm{prox}_{\alpha_k f_i}(v_i(k))\|^2 < \infty$ and $\sum_{k=0}^{\infty} \sum_{i=1}^{m} \mathrm{d}(v_i(k), X)^2 < \infty$ almost surely. This completes the proof. □

The following lemma can be proven by using Lemma 3.2 and by referring to the proof of [24, Lemma 7].

Lemma 3.3 Suppose that Assumptions 2.1, 2.2, and 2.3 hold and define $\bar{v}(k) := (1/m) \sum_{l=1}^{m} v_l(k)$ for all $k \geq 0$. Then $\sum_{k=0}^{\infty} \|e_i(k)\|^2 := \sum_{k=0}^{\infty} \|x_i(k+1) - v_i(k)\|^2 < \infty$ and $\sum_{k=0}^{\infty} \alpha_k \|v_i(k) - \bar{v}(k)\| < \infty$ almost surely for all $i \in V$.

Next, let us show the following lemma.

Lemma 3.4 Suppose that the assumptions in Theorem 3.1 are satisfied, and define $z_i(k) := P_X(v_i(k))$ for all $i \in V$ and for all $k \geq 0$ and $\bar{z}(k) := (1/m) \sum_{i=1}^{m} z_i(k)$ for all $k \geq 0$. Then the sequence $(\|x_i(k) - x^\star\|)_{k \geq 0}$ converges almost surely for all $i \in V$ and for all $x^\star \in X^\star$ and $\liminf_{k \to \infty} f(\bar{z}(k)) = f^\star$ almost surely.

Proof: Choose $x^\star \in X^\star$ arbitrarily. The convexity of $\|\cdot\|^2$ and Assumption 2.3 imply that $\sum_{i=1}^m \|v_i(k) - x^\star\|^2 \leq \sum_{j=1}^m \|x_j(k) - x^\star\|^2$ $(k \geq 0)$. Lemma 3.1 implies that, for all $k \geq 0$ and for all $\tau, \eta, \mu > 0$,

$$\begin{aligned}\sum_{i=1}^m \|x_i(k+1) - x^\star\|^2 \leq& \sum_{i=1}^m \|x_i(k) - x^\star\|^2 - 2\alpha_k \sum_{i=1}^m (f(z_i(k)) - f_i(x^\star))\\&+ \tau \sum_{i=1}^m \|v_i(k) - z_i(k)\|^2 - \left(1 - \frac{1}{\mu}\right) \sum_{i=1}^m \|x_i(k+1) - v_i(k)\|^2\\&+ (\eta + \mu - 2) \sum_{i=1}^m \left\|v_i(k) - \mathrm{prox}_{\alpha_k f_i}(v_i(k))\right\|^2 + mN(\tau, \eta)\alpha_k^2.\end{aligned}$$

From $z_i(k) \in X$ $(i \in V, k \geq 0)$, the convexity of X ensures that $\bar{z}(k) \in X \subset X_i$ $(i \in V)$. Accordingly, (A3) means that $\|\bar{g}_i(k)\| \leq M_i$ for all $\bar{g}_i(k) \in \partial f_i(\bar{z}(k))$ $(i \in V, k \geq 0)$. The definition of ∂f_i $(i \in V)$ and the Cauchy-Schwarz inequality thus guarantee that, for all $i \in V$ and for all $k \geq 0$, $f_i(z_i(k)) - f_i(x^\star) = f_i(z_i(k)) - f_i(\bar{z}(k)) + f_i(\bar{z}(k)) - f_i(x^\star) \geq \langle z_i(k) - \bar{z}(k), \bar{g}_i(k)\rangle + f_i(\bar{z}(k)) - f_i(x^\star) \geq -\bar{M}\|z_i(k) - \bar{z}(k)\| + f_i(\bar{z}(k)) - f_i(x^\star)$, where $\bar{M} := \max_{i\in V} M_i < \infty$. Moreover, the convexity of $\|\cdot\|$ and the nonexpansivity of P_X imply that, for all $i \in V$ and for all $k \geq 0$, $\|z_i(k) - \bar{z}(k)\| = \|(1/m)\sum_{l=1}^m (P_X(v_i(k)) - z_l(k))\| \leq (1/m)\sum_{l=1}^m \|P_X(v_i(k)) - P_X(v_l(k))\| \leq (1/m)\sum_{l=1}^m \|v_i(k) - v_l(k)\|$, which, together with the triangle inequality, implies that, for all $i \in V$ and for all $k \geq 0$, $\|z_i(k) - \bar{z}(k)\| \leq \|v_i(k) - \bar{v}(k)\| + (1/m)\sum_{l=1}^m \|v_l(k) - \bar{v}(k)\|$, where $\bar{v}(k) := (1/m)\sum_{l=1}^m v_l(k)$ $(k \geq 0)$. Accordingly, for all $i \in V$ and for all $k \geq 0$,

$$\begin{aligned}f_i(z_i(k)) - f_i(x^\star) \geq& -\bar{M}\|v_i(k) - \bar{v}(k)\| - \frac{\bar{M}}{m}\sum_{l=1}^m \|v_l(k) - \bar{v}(k)\|\\&+ f_i(\bar{z}(k)) - f_i(x^\star).\end{aligned} \tag{6}$$

Hence, the definitions of f and $f^\star$ imply that, for all $k \geq 0$ and for all $\tau, \eta, \mu > 0$,

$$\begin{aligned}\sum_{i=1}^m \|x_i(k+1) - x^\star\|^2 \leq& \sum_{i=1}^m \|x_i(k) - x^\star\|^2 + 2\bar{M}\alpha_k \sum_{i=1}^m \|v_i(k) - \bar{v}(k)\|\\&+ 2\bar{M}\alpha_k \sum_{l=1}^m \|v_l(k) - \bar{v}(k)\| - 2\alpha_k (f(\bar{z}(k)) - f^\star)\\&+ \tau \sum_{i=1}^m \|v_i(k) - z_i(k)\|^2 - \left(1 - \frac{1}{\mu}\right) \sum_{i=1}^m \|x_i(k+1) - v_i(k)\|^2\\&+ (\eta + \mu - 2) \sum_{i=1}^m \left\|v_i(k) - \mathrm{prox}_{\alpha_k f_i}(v_i(k))\right\|^2 + mN(\tau, \eta)\alpha_k^2,\end{aligned}$$

which, together with $\mathrm{d}(v_i(k), X) = \|v_i(k) - z_i(k)\|$ and $\mathrm{d}(v_i(k), X_i^{\Omega_i(k)}) \leq \|v_i(k) - x_i(k+1)\|$ $(i \in V, k \geq 0)$, implies that, for all $k \geq 0$, for all $\tau, \eta > 0$, and for all

$\mu > 1$,

$$\begin{aligned}\sum_{i=1}^{m} \|x_i(k+1) - x^\star\|^2 \leq \sum_{i=1}^{m} \|x_i(k) - x^\star\|^2 + 4\bar{M}\alpha_k \sum_{i=1}^{m} \|v_i(k) - \bar{v}(k)\| \\ - 2\alpha_k (f(\bar{z}(k)) - f^\star) + \tau \sum_{i=1}^{m} \mathrm{d}(v_i(k), X)^2 \\ - \left(1 - \frac{1}{\mu}\right) \sum_{i=1}^{m} \mathrm{d}\left(v_i(k), X_i^{\Omega_i(k)}\right)^2 \\ + (\eta + \mu - 2) \sum_{i=1}^{m} \left\| v_i(k) - \mathrm{prox}_{\alpha_k f_i}(v_i(k)) \right\|^2 + mN(\tau, \eta)\alpha_k^2.\end{aligned}$$

Accordingly, Assumption 2.5 guarantees that, almost surely, for all $k \geq 0$, for all $\tau, \eta > 0$, and for all $\mu > 1$,

$$\begin{aligned}\mathrm{E}\left[\sum_{i=1}^{m} \|x_i(k+1) - x^\star\|^2 \middle| \mathcal{F}_k\right] \leq \sum_{i=1}^{m} \|x_i(k) - x^\star\|^2 + 4\bar{M}\alpha_k \sum_{i=1}^{m} \|v_i(k) - \bar{v}(k)\| \\ - 2\alpha_k (f(\bar{z}(k)) - f^\star) \\ + \left\{\tau - \frac{1}{c}\left(1 - \frac{1}{\mu}\right)\right\} \sum_{i=1}^{m} \mathrm{d}(v_i(k), X)^2 \\ + (\eta + \mu - 2) \sum_{i=1}^{m} \left\| v_i(k) - \mathrm{prox}_{\alpha_k f_i}(v_i(k)) \right\|^2 \\ + mN(\tau, \eta)\alpha_k^2. \quad (7)\end{aligned}$$

Setting $\tau := 1/(6c)$, $\eta := 1/3$, and $\mu := 3/2$ (also see proof of Lemma 3.2) in the inequality above leads to the finding that, almost surely, for all $k \geq 0$,

$$\begin{aligned}\mathrm{E}\left[\sum_{i=1}^{m} \|x_i(k+1) - x^\star\|^2 \middle| \mathcal{F}_k\right] \leq \sum_{i=1}^{m} \|x_i(k) - x^\star\|^2 - 2\alpha_k (f(\bar{z}(k)) - f^\star) \\ + 4\bar{M}\alpha_k \sum_{i=1}^{m} \|v_i(k) - \bar{v}(k)\| + mN\left(\frac{1}{6c}, \frac{1}{3}\right)\alpha_k^2. \quad (8)\end{aligned}$$

Therefore, since $\bar{z}(k) \in X$ implies $f(\bar{z}(k)) - f^\star \geq 0$ $(k \geq 0)$, Proposition 2.2, (C2), and Lemma 3.3 ensure that $(\|x_i(k) - x^\star\|)_{k \geq 0}$ converges almost surely for all $i \in V$. Moreover,

$$\sum_{k=0}^{\infty} \alpha_k (f(\bar{z}(k)) - f^\star) < \infty \quad (9)$$

is almost surely satisfied. Now, let us assume that $\liminf_{k\to\infty} f(\bar{z}(k)) \leq f^\star$ does not hold almost surely. Then for all $\bar{\Omega} \in \mathcal{F}$, $\Pr\{\bar{\Omega}\} = 1$, and there exists $\omega \in \bar{\Omega}$ such that $\liminf_{k\to\infty} f(\bar{z}(k)(\omega)) - f^\star > 0$. Hence, $k_1 > 0$ and $\gamma > 0$ can be chosen such that $f(\bar{z}(k)(\omega)) - f^\star \geq \gamma$ for all $k \geq k_1$. Accordingly, (9) and (C1) mean that

$$\infty = \gamma \sum_{k=k_1}^{\infty} \alpha_k \leq \sum_{k=k_1}^{\infty} \alpha_k \left(f\left(\bar{z}\left(k\right)\left(\omega\right)\right) - f^\star \right) < \infty,$$

which is a contradiction. Therefore, $\liminf_{k\to\infty} f(\bar{z}(k)) \leq f^\star$ almost surely. From $f(\bar{z}(k)) - f^\star \geq 0$ $(k \geq 0)$, $\liminf_{k\to\infty} f(\bar{z}(k)) = f^\star$ almost surely. This completes the proof. □

It is now possible to prove Theorem 3.1.

Proof: Lemma 3.4 guarantees the almost sure convergence of $(x_i(k))_{k\geq 0}$ $(i \in V)$. From (3), $(v_i(k))_{k\geq 0}$ $(i \in V)$ also converges almost surely. The definition of $\bar{v}(k)$ $(k \geq 0)$ implies the almost sure convergence of $(\bar{v}(k))_{k\geq 0}$. Moreover, Lemma 3.2 implies that, for all $i \in V$, $\sum_{k=0}^{\infty} \mathrm{d}(v_i(k), X)^2 = \sum_{k=0}^{\infty} \|v_i(k) - z_i(k)\|^2 < \infty$ almost surely; i.e., $\lim_{k\to\infty} \|v_i(k) - z_i(k)\| = 0$ almost surely. Accordingly, $(z_i(k))_{k\geq 0}$ $(i \in V)$ converges almost surely. This implies that $(\bar{z}(k))_{k\geq 0}$ converges almost surely to x^*; i.e., $(\bar{z}(k))_{k\geq 0}$ converges in law to x^*. Hence, the closedness of X and Proposition 2.3 guarantee that $\limsup_{k\to\infty} \Pr\{\bar{z}(k) \in X\} \leq \Pr\{x^* \in X\}$. Since the definition of $\bar{z}(k)$ $(k \geq 0)$ implies that $\Pr\{\bar{z}(k) \in X\} = 1$ $(k \geq 0)$, $\Pr\{x^* \in X\} = 1$. Moreover, Lemma 3.4 and the continuity of f ensure that, almost surely,

$$f\left(x^*\right) = \lim_{k\to\infty} f\left(\bar{z}(k)\right) = \liminf_{k\to\infty} f\left(\bar{z}(k)\right) = f^\star; \text{ i.e., } x^* \in X^\star.$$

The definitions of $\bar{v}(k)$ and $\bar{z}(k)$ $(k \geq 0)$ mean that, for all $k \geq 0$, $\|\bar{v}(k) - \bar{z}(k)\| \leq (1/m) \sum_{i=1}^{m} \|z_i(k) - v_i(k)\|$, which, together with $\lim_{k\to\infty} \|v_i(k) - z_i(k)\| = 0$ almost surely for all $i \in V$, implies that $\lim_{k\to\infty} \|\bar{v}(k) - \bar{z}(k)\| = 0$ almost surely. Accordingly, the almost sure convergence of $(\bar{z}(k))_{k\geq 0}$ to $x^* \in X^\star$ guarantees that $(\bar{v}(k))_{k\geq 0}$ also converges almost surely to the same $x^* \in X^\star$.

Since Lemma 3.3 implies that $\sum_{k=0}^{\infty} \alpha_k \|v_i(k) - \bar{v}(k)\| < \infty$ almost surely for all $i \in V$, a discussion similar to the one for obtaining $\liminf_{k\to\infty} f(\bar{z}(k)) \leq f^\star$ almost surely (see proof of Lemma 3.4) and (C1) guarantee that $\liminf_{k\to\infty} \|v_i(k) - \bar{v}(k)\| = 0$ almost surely for all $i \in V$. Moreover, the triangle inequality implies that $\|v_i(k) - x^*\| \leq \|v_i(k) - \bar{v}(k)\| + \|\bar{v}(k) - x^*\|$ $(i \in V, k \geq 0)$. Hence, from $\lim_{k\to\infty} \|\bar{v}(k) - x^*\| = 0$ and $\liminf_{k\to\infty} \|v_i(k) - \bar{v}(k)\| = 0$ $(i \in V)$ almost surely, $\liminf_{k\to\infty} \|v_i(k) - x^*\| = 0$ almost surely for all $i \in V$. Therefore, the almost sure convergence of $(v_i(k))_{k\geq 0}$ $(i \in V)$ leads to the finding that, for all $i \in V$,

$$\lim_{k\to\infty} \|v_i\left(k\right) - x^*\| = 0 \text{ almost surely.} \tag{10}$$

Since Lemma 3.3 ensures that, for all $i \in V$, $\lim_{k\to\infty} \|e_i(k)\|^2 = \lim_{k\to\infty} \|x_i(k+1) - v_i(k)\|^2 = 0$ almost surely, $(x_i(k))_{k\geq 0}$ $(i \in V)$ converges almost surely to $x^* \in X^\star$. This completes the proof. □

3.2. *Convergence rate analysis for Algorithm 3.1*

The discussion in Subsection 3.1 leads to the finding that the sequence of the feasibility error $(\mathrm{d}(x_i(k), X)^2)_{k\geq 0}$ and the sequence of the iteration error $(\|x_i(k) - x^\star\|^2)_{k\geq 0}$ are stochastically decreasing in the sense of the inequalities in the following proposition.

Proposition 3.1 Suppose that the assumptions in Theorem 3.1 hold, that $x^\star \in X^\star$ is a solution to Problem 2.1, and that $(x_i(k))_{k\geq 0}$ $(i \in V)$ is the sequence generated by Algorithm 3.1. Then there exist $\beta^{(j)} > 0$ $(j = 1,2,3,4,5)$ such that, almost surely, for all $k \geq 0$,

$$\mathrm{E}\left[\sum_{i=1}^{m} \mathrm{d}\left(x_i\left(k+1\right), X\right)^2 \middle| \mathcal{F}_k\right] \leq \sum_{i=1}^{m} \mathrm{d}\left(x_i\left(k\right), X\right)^2 - \sum_{j=1,2} \beta^{(j)}\gamma_k^{(j)} + \beta^{(3)}\gamma_k^{(3)},$$

$$\mathrm{E}\left[\sum_{i=1}^{m} \left\|x_i\left(k+1\right) - x^\star\right\|^2 \middle| \mathcal{F}_k\right] \leq \sum_{i=1}^{m} \left\|x_i\left(k\right) - x^\star\right\|^2 - \sum_{j=1,2,5} \beta^{(j)}\gamma_k^{(j)} + \sum_{j=3,4} \beta^{(j)}\gamma_k^{(j)},$$

where $\gamma_k^{(1)} := \sum_{i=1}^m \mathrm{d}(v_i(k), X)^2$, $\gamma_k^{(2)} := \sum_{i=1}^m \|v_i(k) - \mathrm{prox}_{\alpha_k f_i}(v_i(k))\|^2$, $\gamma_k^{(3)} := \alpha_k^2$, $\gamma_k^{(4)} := \alpha_k \sum_{i=1}^m \|v_i(k) - \bar{v}(k)\|$, and $\gamma_k^{(5)} := \alpha_k(f(\bar{z}(k)) - f^\star)$ $(k \geq 0)$ satisfy $\sum_{k=0}^\infty \gamma_k^{(j)} < \infty$ $(j = 1,2,3,4,5)$.

Proof: Lemma 3.2 guarantees that $\gamma_k^{(1)} := \sum_{i=1}^m \mathrm{d}(v_i(k), X)^2$ and $\gamma_k^{(2)} := \sum_{i=1}^m \|v_i(k) - \mathrm{prox}_{\alpha_k f_i}(v_i(k))\|^2$ $(k \geq 0)$ satisfy $\sum_{k=0}^\infty \gamma_k^{(j)} < \infty$ almost surely for $j = 1, 2$. Condition (C2) implies that $\gamma_k^{(3)} := \alpha_k^2$ $(k \geq 0)$ satisfies $\sum_{k=0}^\infty \gamma_k^{(3)} < \infty$. From Lemma 3.3 and (9), $\gamma_k^{(4)} := \alpha_k \sum_{i=1}^m \|v_i(k) - \bar{v}(k)\|$ and $\gamma_k^{(5)} := \alpha_k(f(\bar{z}(k)) - f^\star)$ $(k \geq 0)$ also satisfy $\sum_{k=0}^\infty \gamma_k^{(j)} < \infty$ almost surely for $j = 4, 5$. Set $\tau := 1/(6c), \eta := 1/3$, and $\mu := 3/2$, and put $\beta^{(1)} := -(\tau - (1/c)(1-1/\mu)) = 1/(6c)$, $\beta^{(2)} := 2-\eta-\mu = 1/6$, $\beta^{(3)} := mN(\tau, \eta)$, $\beta^{(4)} := 4\bar{M}$, and $\beta^{(5)} := 2$, where $\beta^{(3)}, \beta^{(4)} < \infty$ hold from (A3) and $\bar{M} := \max_{i \in V} M_i$. Accordingly, (5) and (7) ensure that Proposition 3.1 holds. □

From $-\sum_{j=1,2} \beta^{(j)}\gamma_k^{(j)} + \beta^{(3)}\gamma_k^{(3)} \leq \beta^{(3)}\gamma_k^{(3)} = \beta^{(3)}\alpha_k^2$ and $-\sum_{j=1,2,5} \beta^{(j)}\gamma_k^{(j)} + \sum_{j=3,4} \beta^{(j)}\gamma_k^{(j)} \leq \sum_{j=3,4} \beta^{(j)}\gamma_k^{(j)} = (\beta^{(3)}\alpha_k + \beta^{(4)} \sum_{i=1}^m \|v_i(k) - \bar{v}(k)\|)\alpha_k$ $(k \geq 0)$, Proposition 3.1 and Theorem 3.1 (see (10) for the almost sure boundedness of $(v_i(k))_{k\geq 0}$ $(i \in V)$) indicate that $(x_i(k))_{k \geq 0}$ $(i \in V)$ in Algorithm 3.1 converges almost surely to a solution to Problem 2.1 under the following convergence rate conditions: for all $k \geq 0$,

$$\begin{aligned}
&\mathrm{E}\left[\sum_{i=1}^{m} \mathrm{d}\left(x_i\left(k+1\right), X\right)^2 \middle| \mathcal{F}_k\right] \leq \sum_{i=1}^{m} \mathrm{d}\left(x_i\left(k\right), X\right)^2 + O\left(\alpha_k^2\right), \\
&\mathrm{E}\left[\sum_{i=1}^{m} \left\|x_i\left(k+1\right) - x^\star\right\|^2 \middle| \mathcal{F}_k\right] \leq \sum_{i=1}^{m} \left\|x_i\left(k\right) - x^\star\right\|^2 + O\left(\alpha_k\right).
\end{aligned} \tag{11}$$

Moreover, under the condition that $\|x_i(k+1) - x^\star\| \approx \|x_i(k) - x^\star\|$ holds for all $i \in V$ and for a large enough k, (8) means that, almost surely,

$$f\left(\bar{z}(k)\right) \approx f^\star + 2\bar{M} \sum_{i=1}^{m} \left\|v_i(k) - \bar{v}(k)\right\| + O\left(\alpha_k\right), \tag{12}$$

where it is guaranteed from Theorem 3.1 that $(\|v_i(k) - \bar{v}(k)\|)_{k \geq 0}$ $(i \in V)$ converges almost surely to 0.

From Proposition 3.1, the convergence rate of Algorithm 3.1 depends on $\beta^{(j)}$ and $(\gamma_k^{(j)})_{k\geq 0}$ $(j = 1,2,3,4,5)$; i.e., the number of users m and the step size $(\alpha_k)_{k\geq 0}$

(see (11) and (12)). When m is fixed, it is desirable to set $(\alpha_k)_{k\geq 0}$ so that, for all $k \geq 0$, $\mathrm{E}[\sum_{i=1}^m \mathrm{d}(x_i(k+1), X)^2 | \mathcal{F}_k] < \sum_{i=1}^m \mathrm{d}(x_i(k), X)^2$ and $\mathrm{E}[\sum_{i=1}^m \|x_i(k+1) - x^\star\|^2 | \mathcal{F}_k] < \sum_{i=1}^m \|x_i(k) - x^\star\|^2$ are almost surely satisfied. Accordingly, Algorithm 3.1converges quickly if $(\alpha_k)_{k\geq 0}$ is chosen so as to satisfy

$$-\sum_{j=1,2} \beta^{(j)} \gamma_k^{(j)} + \beta^{(3)} \gamma_k^{(3)} < 0 \text{ and } -\sum_{j=1,2,5} \beta^{(j)} \gamma_k^{(j)} + \sum_{j=3,4} \beta^{(j)} \gamma_k^{(j)} < 0 \ (k \geq 0).$$

Hence, it would be desirable to set $(\alpha_k)_{k\geq 0}$ so as to satisfy $\sum_{j=3,4} \beta^{(j)} \gamma_k^{(j)} = \beta^{(3)} \alpha_k^2 + \beta^{(4)} \alpha_k \sum_{i=1}^m \|v_i(k) - \bar{v}(k)\| \approx 0$ as much as possible, e.g., to set $\alpha_k := \alpha/(k+1)$ with a small positive constant α. Section 5 gives numerical examples showing that Algorithm 3.1 with $\alpha_k := 10^{-3}/(k+1)$ converges more quickly than Algorithm 3.1 with $\alpha_k := 1/(k+1)$.

Here, let us compare the convergence analysis of Algorithm 3.1 (Theorem 3.1 and Proposition 3.1) with the convergence analysis of the incremental subgradient-proximal methods in [4, Section 4]. The problem considered in [4] was

$$\text{minimize } F(x) = \sum_{i \in V} F_i(x) := \sum_{i \in V} \left(f_i(x) + h_i(x) \right) \text{ subject to } x \in X, \tag{13}$$

where $f_i, h_i \colon \mathbb{R}^d \to \mathbb{R}$ $(i \in V)$ are convex and $X \subset \mathbb{R}^d$ is nonempty, closed, and convex. Under the conditions that f_i $(i \in V)$ are suitable for a proximal iteration and the subgradients of h_i $(i \in V)$ are efficiently computed, the following method [4, Section 4, (4.2)] was proposed for solving Problem (13):

$$\begin{aligned} &z(k) := \mathrm{prox}_{\alpha_k f_{\omega_k}} (x(k)), \ g_{\omega_k} \in \partial h_{\omega_k} (z(k)), \\ &x(k+1) := P_X \left(z(k) - \alpha_k g_{\omega_k} \right), \end{aligned} \tag{14}$$

where $(\omega_k)_{k\geq 0} \subset V := \{1, 2, \ldots, m\}$ is a sequence of random variables.

Algorithm (14) is an incremental optimization algorithm that uses the metric projection onto the whole constraint set X and the proximity operator and subdifferential of one function selected randomly from $\{F_i\}_{i \in V}$ while Algorithm 3.1 is a distributed optimization algorithm that uses the proximity operator of each user's objective function and the metric projection onto one closed convex set selected randomly from each user's constraint sets. Proposition 4.3 in [4] shows that, under certain assumptions, Algorithm (14) with $(\alpha_k)_{k\geq 0}$ satisfying (C1) and (C2) converges almost surely to some random point in the solution set of Problem (13). Theorem 3.1 guarantees the almost sure convergence of Algorithm 3.1 with the conditions (C1) and (C2) to a random point in $X^\star$. Proposition 4.2 in [4] indicates that, under certain assumptions, Algorithm (14) with $\alpha_k := \alpha > 0$ $(k \geq 0)$ satisfies that, for all $\epsilon > 0$, almost surely

$$\min_{0 \leq k \leq N} F\left(x(k)\right) \leq F^\star + \frac{5 \alpha m c^2 + \epsilon}{2}, \tag{15}$$

where $F^\star$ stands for the optimal value of Problem (13), $c \in \mathbb{R}$ is a constant, $\bar{X}^\star$ is the solution set of Problem (13), and N is a random variable with $\mathrm{E}[N] \leq m \mathrm{d}(x(0), \bar{X}^\star)^2/(\alpha \epsilon)$. From (15) and Proposition 3.1, the convergence rates of Algorithms 3.1 and (14) depend on the number of elements in V and the step size $(\alpha_k)_{k\geq 0}$.

4. Distributed random projected subgradient algorithm

This section presents the subgradient algorithm with random projections for solving Problem 2.1.

Algorithm 4.1

Step 0. User i ($i \in V$) sets $x_i(0)$ arbitrarily.
Step 1. User i ($i \in V$) receives $x_j(k)$ from its neighboring users $j \in N_i(k)$ and computes the weighted average $v_i(k)$ defined as in (3) and the subgradient $g_i(k) \in \partial f_i(v_i(k))$. User i updates its estimate $x_i(k+1)$ by

$$x_i\left(k+1\right) := P_{X_i^{\Omega_i(k)}}\left(v_i\left(k\right) - \alpha_k g_i\left(k\right)\right). \tag{16}$$

The algorithm sets $k := k+1$ and returns to Step 1.

In this section, Assumption (A3) is replaced with

(A3)' For all $i \in V$, there exists $M_i \in \mathbb{R}$ such that $\sup\{\|g_i\| : x \in X_i, g_i \in \partial f_i(x)\} \leq M_i$. For all $i \in V$, there exists $C_i \in \mathbb{R}$ such that $\sup\{\|g_i(k)\| : g_i(k) \in \partial f_i(v_i(k)), k \geq 0\} \leq C_i$.

It is obvious that Assumption (A3)' is stronger than Assumption (A3). Assumption (A3)' holds when f_i ($i \in V$) is Lipschitz continuous on $\mathbb{R}^d$ (Proposition 2.1(iv)). The boundedness of $(g_i(k))_{k\geq 0}$ ($i \in V$) is needed to show that Algorithm 4.1 satisfies $\sum_{k=0}^{\infty} \mathrm{d}(v_i(k), X)^2 < \infty$ almost surely for all $i \in V$, which is the essential part of the convergence analysis of Algorithm 4.1 (see Lemmas 4.1 and 4.2 for details).

Let us do a convergence analysis of Algorithm 4.1.

Theorem 4.1 Under Assumptions 2.1–2.3, the sequence $(x_i(k))_{k\geq 0}$ ($i \in V$) generated by Algorithm 4.1 converges almost surely to a random point $x^\star \in X^\star$.

Let us compare the distributed random projected gradient algorithm [24, (2a) and (2b)] with Algorithm 4.1. The algorithm [24, (2a) and (2b)] is the pioneering distributed optimization algorithm that is based on local communications of users' estimates in a network and a gradient descent with random projections. It can be applied to Problem 2.1 when f_i ($i \in V$) is convex and differentiable with the Lipschitz gradient ∇f_i [24, Assumption 1 c)]. The algorithm [24, (2a) and (2b)] is

$$x_i\left(k+1\right) := P_{X_i^{\Omega_i(k)}}\left(v_i\left(k\right) - \alpha_k \nabla f_i\left(v_i\left(k\right)\right)\right), \tag{17}$$

where $v_i(k)$ is defined as in (3). Proposition 1 in [24] indicates that, under the assumptions in Theorem 3.1, the sequence $(x_i(k))_{k\geq 0}$ ($i \in V$) generated by Algorithm (17) converges almost surely to $x^\star \in X^\star$. In contrast to Algorithm (17), Algorithm 4.1 can be applied to nonsmooth convex optimization (see Assumption 2.1(A2)) by using the subgradients $g_i(k) \in \partial f_i(v_i(k))$ ($i \in V, k \geq 0$) and enables all users to arrive at the same solution to Problem 2.1 under the assumptions in Theorem 4.1 that are stronger than the ones in Theorem 3.1. In Algorithm 3.1, each user sets $x_i(k+1)$ by using its proximity operator (see (4) for definition of $x_i(k)$ ($k \geq 0$) in Algorithm 3.1) and can solve Problem 2.1 under the assumptions in Theorem 3.1 (see Theorem 3.1).

4.1. *Proof of Theorem 4.1*

The proof starts with the following lemma, which can be proven by referring to the proof of [26, Lemma 2].

Lemma 4.1 Suppose that Assumption 2.1 holds. Then for all $x \in X$, for all $i \in V$, for all $z \in X_i$, for all $k \geq 0$, and for all $\tau, \eta > 0$,

$$\begin{aligned} \|x_i(k+1) - x\|^2 \leq \|v_i(k) - x\|^2 - 2\alpha_k (f_i(z) - f_i(x)) + M(\tau, \eta)\alpha_k^2 \\ + \tau \|v_i(k) - z\|^2 + (\eta - 1)\|v_i(k) - x_i(k+1)\|^2, \end{aligned}$$

where $M(\tau, \eta) := \max_{i \in V}(M_i^2/\tau + C_i^2/\eta) < \infty$, and M_i and C_i $(i \in V)$ are as in Assumption (A3)'.

The following lemma can also be proven by referring to the proof of Lemma 3.2 and by using Lemma 4.1 (see [20] for the details of the proof of Lemma 4.2).

Lemma 4.2 Suppose that Assumptions 2.1, 2.5, and 2.3 hold. Then $\sum_{k=0}^{\infty} \mathrm{d}(v_i(k), X)^2 < \infty$ almost surely for all $i \in V$.

Proof: A discussion similar to the one for obtaining (5) implies that, almost surely, for all $k \geq 0$,

$$\begin{aligned} \mathrm{E}\left[\sum_{i=1}^{m} \mathrm{d}(x_i(k+1), X)^2 \middle| \mathcal{F}_k\right] \leq \sum_{j=1}^{m} \mathrm{d}(x_j(k), X)^2 - \frac{1}{4c}\sum_{i=1}^{m} \mathrm{d}(v_i(k), X)^2 \\ + mM\left(\frac{1}{2c}, \frac{1}{4}\right)\alpha_k^2. \end{aligned} \tag{18}$$

Proposition 2.2 and (C2) lead to $\sum_{k=0}^{\infty}\sum_{i=1}^{m} \mathrm{d}(v_i(k), X)^2 < \infty$ almost surely; i.e., $\sum_{k=0}^{\infty} \mathrm{d}(v_i(k), X)^2 < \infty$ almost surely for all $i \in V$. This completes the proof. □

A discussion similar to the one for proving Lemma 3.3 leads to the following (The proof of Lemma 4.3 is also described in [20]).

Lemma 4.3 Suppose that Assumptions 2.1, 2.2, and 2.3 hold, and define $\bar{v}(k) := (1/m)\sum_{l=1}^{m} v_l(k)$ and $e_i(k) := x_i(k+1) - v_i(k)$ for all $k \geq 0$ and for all $i \in V$. Then $\sum_{k=0}^{\infty} \|e_i(k)\|^2 < \infty$ and $\sum_{k=0}^{\infty} \alpha_k \|v_i(k) - \bar{v}(k)\| < \infty$ almost surely for all $i \in V$.

The same discussion as for the proof of Lemma 3.4 leads to the following (see [20] for the details of the proof of Lemma 4.4).

Lemma 4.4 Suppose that the assumptions in Theorem 4.1 are satisfied and define $z_i(k) := P_X(v_i(k))$ for all $i \in V$ and for all $k \geq 0$ and define $\bar{z}(k) := (1/m)\sum_{i=1}^{m} z_i(k)$ for all $k \geq 0$. Then the sequence $(\|x_i(k) - x^\star\|)_{k \geq 0}$ converges almost surely for all $i \in V$ and for all $x^\star \in X^\star$, and $\liminf_{k \to \infty} f(\bar{z}(k)) = f^\star$ almost surely.

Proof: Choose $x^\star \in X^\star$ arbitrarily. It can be observed that (6) holds for Algorithm 4.1 because the definitions of $\bar{z}(k)$ and $\bar{v}(k)$ $(k \geq 0)$ in the proof of Theorem 3.1 are the same as the definitions of $\bar{z}(k)$ and $\bar{v}(k)$ $(k \geq 0)$ in Lemma 4.4. A discussion similar to the one for proving (7) means that, almost surely for all $k \geq 0$, for all

$\tau > 0$, and for all $\eta \in (0,1)$,

$$\mathrm{E}\left[\sum_{i=1}^{m}\|x_i(k+1)-x^\star\|^2 \middle| \mathcal{F}_k\right] \leq \sum_{i=1}^{m}\|x_i(k)-x^\star\|^2 + 4\bar{M}\alpha_k\sum_{i=1}^{m}\|v_i(k)-\bar{v}(k)\| - 2\alpha_k\left(f(\bar{z}(k))-f^\star\right) + \tau\sum_{i=1}^{m}\mathrm{d}\left(v_i(k),X\right)^2 + \frac{\eta-1}{c}\sum_{i=1}^{m}\mathrm{d}\left(v_i(k),X\right)^2 + mM(\tau,\eta)\,\alpha_k^2, \quad (19)$$

which, together with $\tau := 1/(2c)$, $\eta := 1/4$, and $\tau + (\eta-1)/c = -1/(4c)$, implies that, almost surely, for all $k \geq 0$,

$$\mathrm{E}\left[\sum_{i=1}^{m}\|x_i(k+1)-x^\star\|^2 \middle| \mathcal{F}_k\right] \leq \sum_{i=1}^{m}\|x_i(k)-x^\star\|^2 - 2\alpha_k\left(f(\bar{z}(k))-f^\star\right) + 4\bar{M}\alpha_k\sum_{i=1}^{m}\|v_i(k)-\bar{v}(k)\| + mM\left(\frac{1}{2c},\frac{1}{4}\right)\alpha_k^2, \quad (20)$$

where $M(1/(2c),1/4) < \infty$ holds from (A3)'. Therefore, Proposition 2.2, (C2), and Lemma 4.3 ensure that $(\sum_{i=1}^m \|x_i(k)-x^\star\|^2)_{k\geq 0}$ converges almost surely; i.e., $(\|x_i(k)-x^\star\|)_{k\geq 0}$ converges almost surely for all $i \in V$. Moreover, since $\bar{z}(k) \in X$ implies $f(\bar{z}(k)) - f^\star \geq 0$ $(k \geq 0)$, there is also another finding: almost surely

$$\sum_{k=0}^{\infty}\alpha_k\left(f(\bar{z}(k))-f^\star\right) < \infty. \quad (21)$$

Hence, a discussion similar to the one for proving Lemma 3.4 leads to $\liminf_{k\to\infty} f(\bar{z}(k)) = f^\star$ almost surely. This completes the proof. □

Theorem 4.1 can be proven by referring to the proof of Theorem 3.1.

Proof: By referring to the proof of Theorem 3.1, the almost sure convergence of $(x_i(k))_{k\geq 0}$ $(i \in V)$ and Lemma 4.2 lead to the conclusion that $(v_i(k))_{k\geq 0}$, $(z_i(k))_{k\geq 0}$ $(i \in V)$, $(\bar{v}(k))_{k\geq 0}$, and $(\bar{z}(k))_{k\geq 0}$ converge almost surely. Moreover, Lemma 4.4 and the continuity of f ensure that $(\bar{v}(k))_{k\geq 0}$ and $(\bar{z}(k))_{k\geq 0}$ converge almost surely to $x_* \in X^\star$. By referring to the proof of Theorem 3.1, Lemma 4.3, (C1), and the triangle inequality guarantee that $\lim_{k\to\infty}\|v_i(k)-x_*\| = 0$ almost surely for all $i \in V$. Since Lemma 4.3 ensures that, for all $i \in V$, $\lim_{k\to\infty}\|e_i(k)\|^2 = \lim_{k\to\infty}\|x_i(k+1)-v_i(k)\|^2 = 0$ almost surely, $(x_i(k))_{k\geq 0}$ $(i \in V)$ converges almost surely to $x_* \in X^\star$. This completes the proof. □

4.2. *Convergence rate analysis for Algorithm 4.1*

The following proposition is proven by referring to the discussion in Subsection 4.1.

Proposition 4.1 Suppose that the assumptions in Theorem 4.1 hold, that $x^\star \in X^\star$ is a solution to Problem 2.1, and that $(x_i(k))_{k\geq 0}$ $(i \in V)$ is the sequence generated by Algorithm 4.1. Then there exist $\beta^{(j)} > 0$ $(j = 1,2,3,4)$ such that, almost surely, for all $k \geq 0$,

$$\mathrm{E}\left[\sum_{i=1}^{m} \mathrm{d}\left(x_i\left(k+1\right), X\right)^2 \middle| \mathcal{F}_k\right] \leq \sum_{i=1}^{m} \mathrm{d}\left(x_i\left(k\right), X\right)^2 - \beta^{(1)}\gamma_k^{(1)} + \beta^{(2)}\gamma_k^{(2)},$$

$$\mathrm{E}\left[\sum_{i=1}^{m} \left\|x_i\left(k+1\right) - x^\star\right\|^2 \middle| \mathcal{F}_k\right] \leq \sum_{i=1}^{m} \left\|x_i\left(k\right) - x^\star\right\|^2 - \sum_{j=1,4} \beta^{(j)}\gamma_k^{(j)} + \sum_{j=2,3} \beta^{(j)}\gamma_k^{(j)},$$

where $\gamma_k^{(1)} := \sum_{i=1}^m \mathrm{d}(v_i(k), X)^2$, $\gamma_k^{(2)} := \alpha_k^2$, $\gamma_k^{(3)} := \alpha_k \sum_{i=1}^m \|v_i(k) - \bar{v}(k)\|$, and $\gamma_k^{(4)} := \alpha_k(f(\bar{z}(k)) - f^\star)$ $(k \geq 0)$ satisfy $\sum_{k=0}^\infty \gamma_k^{(j)} < \infty$ $(j = 1, 2, 3, 4)$.

Proof: From Lemma 4.2 and (C2), $\gamma_k^{(1)} := \sum_{i=1}^m \mathrm{d}(v_i(k), X)^2$ and $\gamma_k^{(2)} := \alpha_k^2$ $(k \geq 0)$ satisfy $\sum_{k=0}^\infty \gamma_k^{(1)} < \infty$ almost surely and $\sum_{k=0}^\infty \gamma_k^{(2)} < \infty$. Lemma 4.3 and (21) ensure that $\gamma_k^{(3)} := \alpha_k \sum_{i=1}^m \|v_i(k) - \bar{v}(k)\|$ and $\gamma_k^{(4)} := \alpha_k(f(\bar{z}(k)) - f^\star)$ $(k \geq 0)$ also satisfy $\sum_{k=0}^\infty \gamma_k^{(j)} < \infty$ almost surely for $j = 3, 4$. Set $\tau := 1/(2c)$ and $\eta := 1/4$, $\beta^{(1)} := -(\tau + (\eta - 1)/c) = 1/(4c)$, $\beta^{(2)} := mM(\tau, \eta)$, $\beta^{(3)} := 4\bar{M}$, and $\beta^{(4)} := 2$, where $\beta^{(2)}, \beta^{(3)} < \infty$ hold from (A3)' and $\bar{M} := \max_{i \in V} M_i$. Accordingly, (18) and (19) ensure that Proposition 4.1 holds. □

A discussion similar to the one for obtaining (11), Proposition 4.1, and Theorem 4.1 indicates that $(x_i(k))_{k \geq 0}$ $(i \in V)$ in Algorithm 4.1 converges almost surely to a solution to Problem 2.1 for the following convergence rates: for all $k \geq 0$,

$$\begin{aligned}
\mathrm{E}\left[\sum_{i=1}^{m} \mathrm{d}\left(x_i\left(k+1\right), X\right)^2 \middle| \mathcal{F}_k\right] &\leq \sum_{i=1}^{m} \mathrm{d}\left(x_i\left(k\right), X\right)^2 + O\left(\alpha_k^2\right), \\
\mathrm{E}\left[\sum_{i=1}^{m} \left\|x_i\left(k+1\right) - x^\star\right\|^2 \middle| \mathcal{F}_k\right] &\leq \sum_{i=1}^{m} \left\|x_i\left(k\right) - x^\star\right\|^2 + O\left(\alpha_k\right).
\end{aligned} \tag{22}$$

Under the condition that $\|x_i(k+1) - x^\star\| \approx \|x_i(k) - x^\star\|$ for all $i \in V$ and for a large enough k, (20) implies that, almost surely,

$$f\left(\bar{z}(k)\right) \approx f^\star + 2\bar{M} \sum_{i=1}^{m} \|v_i(k) - \bar{v}(k)\| + O(\alpha_k), \tag{23}$$

where $(\|v_i(k) - \bar{v}(k)\|)_{k \geq 0}$ is the almost sure convergent sequence (see proof of Theorem 4.1). It can be observed from (11), (12), (22), and (23) that Algorithms 3.1 and 4.1 have almost the same convergence rate. Section 5 gives numerical examples showing that Algorithms 3.1 and 4.1 have almost the same convergence rate when they have the same step size.

Next, let us consider the case in which f_i $(i \in V)$ is convex and differentiable and ∇f_i $(i \in V)$ satisfies the Lipschitz continuity condition [24, Assumption 1 c)]. Then Algorithm 4.1 coincides with the first distributed random projection algorithm [24, (2a) and (2b)] (see (17)) defined as follows for all $k \geq 0$ and for all $i \in V$:

$$x_i\left(k+1\right) := P_{X_i^{\Omega_i(k)}}\left(v_i\left(k\right) - \alpha_k \nabla f_i\left(v_i\left(k\right)\right)\right), \tag{24}$$

where $v_i(k)$ $(i \in V, k \geq 0)$ is as in (3) and $(\alpha_k)_{k \geq 0}$ satisfies (C1) and (C2). The proof of Lemma 5 and (18) in [24] indicate that algorithm (24), under (A3), almost surely satisfies the following convergence rate conditions for a large enough k.

$$\begin{aligned}
&\mathrm{E}\left[\sum_{i=1}^{m} \mathrm{d}\left(x_i\left(k+1\right), X\right)^2 \middle| \mathcal{F}_k\right] \leq \left(1 + O\left(\alpha_k^2\right)\right) \sum_{i=1}^{m} \mathrm{d}\left(x_i\left(k\right), X\right)^2 + O\left(\alpha_k^2\right),\\
&\mathrm{E}\left[\sum_{i=1}^{m} \left\|x_i\left(k+1\right) - x^\star\right\|^2 \middle| \mathcal{F}_k\right] \leq \left(1 + O\left(\alpha_k^2\right)\right) \sum_{i=1}^{m} \left\|x_i\left(k\right) - x^\star\right\|^2 + O\left(\alpha_k\right).
\end{aligned} \tag{25}$$

Proposition 4.1 implies that, if the stronger assumption (A3)' is satisfied, Algorithm (24) satisfies (22) that are better properties for convergence rate than (25).

Let us compare the convergence analysis used here (Theorem 4.1 and Proposition 4.1) with that of the incremental subgradient methods in [27]. The following incremental subgradient method with randomization can optimize the sum of nonsmooth, convex functions $\sum_{i \in V} f_i$ over a nonempty, closed convex set X [27, (3.1)]:

$$\begin{aligned}
&g_{\omega_k} \in \partial f_{\omega_k}(x(k)),\\
&x(k+1) := P_X\left(x(k) - \alpha_k g_{\omega_k}\right),
\end{aligned} \tag{26}$$

where $(\omega_k)_{k \geq 0} \subset V$ is a sequence of random variables.

Algorithm (26) is an incremental optimization algorithm that uses the metric projection onto the whole constraint set X and the subdifferential of one function selected randomly from $\{f_i\}_{i \in V}$ while Algorithm 4.1 is a distributed optimization algorithm that uses the subdifferential of each user's objective function and the metric projection onto one closed convex set selected randomly from each user's constraint sets. Under certain assumptions, Algorithm (26) with $(\alpha_k)_{k \geq 0}$ satisfying (C1) and (C2) converges almost surely to some minimizer of $\sum_{i \in V} f_i$ over X [27, Proposition 3.2]. Theorem 4.1 with conditions (C1) and (C2) indicates the almost sure convergence of Algorithm 4.1 to a random point in $X^\star$. Proposition 3.1 in [27] indicates that, under certain assumptions, Algorithm (26) with $\alpha_k := \alpha > 0$ $(k \geq 0)$ means that the following relationship is almost surely satisfied.

$$\inf_{k \geq 0} \sum_{i \in V} f_i\left(x(k)\right) \leq \sum_{i \in V} f_i\left(x^\star\right) + \frac{\alpha m c^2}{2}, \tag{27}$$

where $x^\star \in X$ represents the minimizer of $\sum_{i \in V} f_i$ over X and $c \in \mathbb{R}$ is a constant. From (27) and Proposition 4.1, the convergence rates of Algorithms 4.1 and (26) depend on the number of elements in V and the step size $(\alpha_k)_{k \geq 0}$, as seen in Algorithms 3.1 and (14) (see Subsection 3.2).

5. Numerical evaluation

Let us apply Algorithms 3.1 and 4.1 to Problem 2.1 with $f_i \colon \mathbb{R}^d \to \mathbb{R}$, and $X_i \subset \mathbb{R}^d$ $(i \in V := \{1, 2, \ldots, m\})$ defined by

$$f_i\left(x\right) := \sum_{j=1}^{d} a_{ij} \left|x_j - b_{ij}\right| \text{ and } X_i := \bigcap_{j=1}^{d} \left\{x \in \mathbb{R}^d \colon \left\|x - c_{ij}\right\| \leq r_{ij}\right\},$$

where $a_i := (a_{ij})_{j=1}^d, b_i := (b_{ij})_{j=1}^d, r_i := (r_{ij})_{j=1}^d \in \mathbb{R}_+^d$ $(i \in V)$, and $c_{ij} \in \mathbb{R}^d$ $(i \in V, j = 1, 2, \ldots, d)$, meaning that this is the problem of minimizing the sum of the weighted L^1-norms $f(x) = \sum_{i=1}^m f_i(x)$ over the intersection of closed balls $X = \bigcap_{i=1}^m X_i$. The metric projection onto $X_i^j := \{x \in \mathbb{R}^d \colon \|x - c_{ij}\| \leq r_{ij}\}$ $(i \in V, j = 1, 2, \ldots, d)$ can be computed within a finite number of arithmetic operations [1, Chapter 28]. Function f_i $(i \in V)$ satisfies the Lipschitz continuity condition. Hence, Assumptions 2.1 and 2.5 hold. The set $X_i^{\Omega_i(k)}$ $(i \in V, k \geq 0)$ used in the numerical evaluation was chosen randomly from the sets X_i^j so as to satisfy Assumption 2.4. The subdifferential ∂f_i and the proximity operator $\mathrm{prox}_{\alpha f_i}$ $(i \in V, \alpha > 0)$ can be calculated explicitly [13, Lemma 10, (30), (35)].

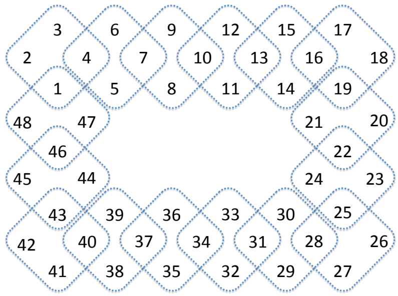


Figure 1.: Network model used in numerical evaluation, where users within each marked area can communicate with each other

In the evaluation, a network with 48 users (i.e., $m := 48$) and 16 subnetworks was used, as illustrated in Figure 1. It was assumed that users within each marked area (i.e., all users in each subnetwork) could communicate with each other. For example, user 2 could communicate with users 1, 3, and 4 (i.e., $N_2(k) = \{1, 2, 3, 4\}$) while user 1 could communicate with not only users 2, 3, and 4 but also users 46, 47, and 48 (i.e., $N_1(k) = \{1, 2, 3, 4, 46, 47, 48\}$). The weighted parameters $w_{ij}(k)$ $(i \in V, j \in N_i(k))$ were set to satisfy Assumption 2.3 (e.g., user 2 had $w_{2,j}$ $(j \in N_2(k))$ such that $w_{2,j}(k) = w_{j,2}(k) = 3/8$ $(j = 2, 3)$ and $w_{2,j}(k) = w_{j,2}(k) = 1/8$ $(j = 1, 4)$, and user 1 had $w_{1,j}$ $(j \in N_1(k))$ such that $w_{1,1}(k) = 2/8$ and $w_{1,j}(k) = w_{j,1} = 1/8$ $(j = 2, 3, 4, 46, 47, 48)$). The point $v_i(k)$ $(i \in V)$ defined in (3) (e.g., $v_2(k) = (3/8)(x_2(k) + x_3(k)) + (1/8)(x_1(k) + x_4(k))$) was computed by passing along $x_j(k)$ $(j \in N_i(k))$ in a prearranged cyclic order.

The computer used in the evaluation had two Intel Xeon E5-2640 v3 (2.60 GHz) CPUs. Each had 8 physical cores and 16 threads; i.e., the total number of cores was 16, and the total number of threads was 32. The computer had 64 GB DDR4 memory and ran the Ubuntu 14.04.1 (Linux kernel: 3.16.0-30-generic, 64 bit) operating system. The evaluation programs were run in Python 3.4.0; Numpy 1.8.2 was used to compute the linear algebra operations. We set $d := 100$ and $m := 48$ and used $a_i := (a_{ij})_{j=1}^d \in (0, 1]^d, b_i := (b_{ij})_{j=1}^d \in [0, 1)^d, r_i := (r_{ij})_{j=1}^d \in [3, 4)^d$ $(i \in V)$, and $c_{ij} \in [-\sqrt{(3/4)d}, \sqrt{(3/4)d})^d$ $(i \in V, j = 1, 2, \ldots, d)$ to satisfy $X \neq \emptyset$ generated randomly by `numpy.random`. One hundred samplings were performed, each starting from different random initial points $x_i(0)$ $(i \in V)$ in the range $[-2, 2]^d$, and the results were averaged.

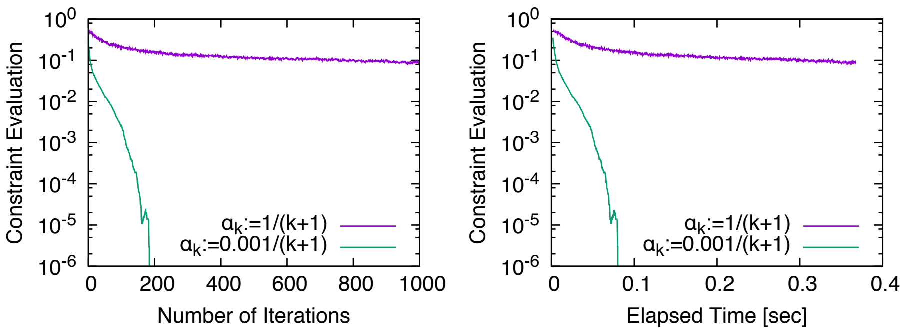
From the discussion in Subsections 3.2 and 4.2, it can be expected that Algorithms 3.1 and 4.1 with small step sizes converge quickly. To see how the step size affects their convergence rates, we compared their rates for $\alpha_k = 1/(k+1)$ with those for $\alpha_k = 10^{-3}/(k+1)$. Step size $\alpha_k = 1/(k+1)$ is the simplest sequence satisfying (C1) and (C2) in Assumption 2.3. The selection of step size $\alpha_k = 10^{-3}/(k+1)$ was based on the numerical results in [16, 22], which presented fixed point optimization algorithms that minimize the sum of smooth, convex functions over the intersection of simple, closed convex sets. The previous results [16, 22] indicated that fixed point optimization algorithms with a small step size such as $\alpha_k = 10^{-3}/(k+1)$ converge more quickly than ones with the standard step size $\alpha_k = 1/(k+1)$. Accordingly, the numerical evaluation described here used step sizes $\alpha_k = 1/(k+1)$ and $10^{-3}/(k+1)$ and investigated which one resulted in quicker convergence for Algorithms 3.1 and 4.1.

Let us define two performance measures for each $i \in V$ and for all $k \geq 0$,

$$D_i(k) := \left\| x_i(k) - \prod_{j=1}^{d} P_{X_i^j}(x_i(k)) \right\| \text{ and } F_i(k) := f_i(x_i(k)), \tag{28}$$

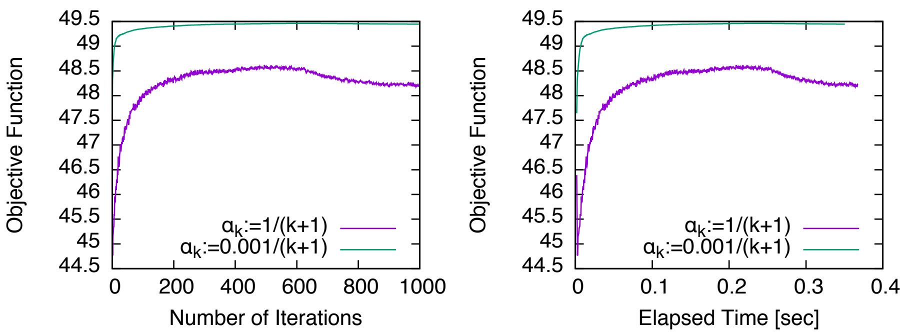
and observe the behaviors of $D_i(k)$ and $F_i(k)$ for Algorithms 3.1 and 4.1 with $\alpha_k = 1/(k+1)$ and $10^{-3}/(k+1)$. If $(D_i(k))_{k\geq 0}$ $(i \in V)$ converges to 0, $(x_i(k))_{k\geq 0}$ converges to a fixed point of $\prod_{j=1}^{d} P_{X_i^j}$, i.e., to a point in $X_i = \bigcap_{j=1}^{d} X_i^j$ [1, Corollary 4.37].

First, let us observe the behaviors of $(x_1(k))_{k=0}^{1000}$ calculated for user 1, which belongs to two subnetworks. Figures 2 and 3 show that Algorithm 3.1 converged more quickly to a point in X_1 with $\alpha_k = 10^{-3}/(k+1)$ than with $\alpha_k = 1/(k+1)$. From Figures 2 and 3, it can be seen that Algorithm 3.1 with $\alpha_k = 10^{-3}/(k+1)$ optimizes f_1 over X_1 in the early stages.



(a) $D_1(k)$ vs. no. of iterations (b) $D_1(k)$ vs. elapsed time

Figure 2.: Behaviors of $D_1(k)$ for Algorithm 3.1 with $\alpha_k = 1/(k+1)$ and $10^{-3}/(k+1)$

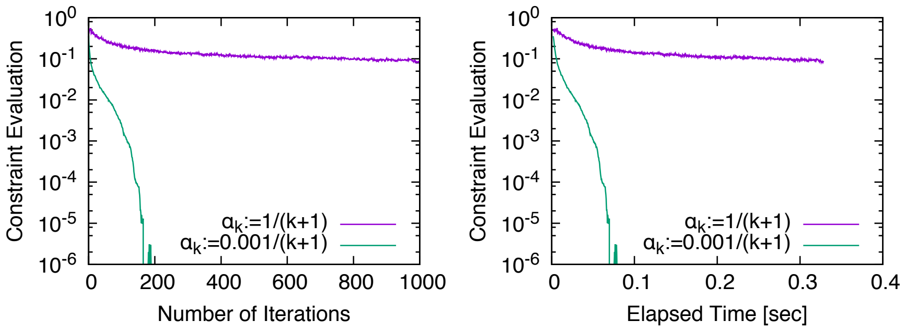


(a) $F_1(k)$ vs. no. of iterations (b) $F_1(k)$ vs. elapsed time

Figure 3.: Behaviors of $F_1(k)$ for Algorithm 3.1 with $\alpha_k = 1/(k+1)$ and $10^{-3}/(k+1)$

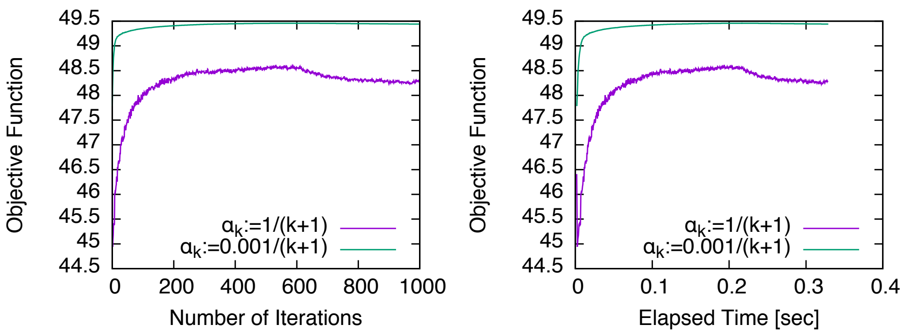
Figures 4 and 5 show that Algorithm 4.1 also converged more quickly to a point in X_1 with $\alpha_k = 10^{-3}/(k+1)$ than with $\alpha_k = 1/(k+1)$. This is consistent with the finding that the algorithms had the same convergence rate (see (11) and (22)).

Investigation of the behaviors of $(D_i(k))_{k=0}^{1000}$ and $(F_i(k))_{k=0}^{1000}$ $(i = 2, 3, \ldots, 48)$ revealed that $(x_i(k))_{k=0}^{1000}$ $(i = 2, 3, \ldots, 48)$ generated by Algorithms 3.1 and 4.1 with $\alpha_k = 10^{-3}/(k+1)$ converged more quickly than $(x_i(k))_{k=0}^{1000}$ generated by Algorithms 3.1 and 4.1 with $\alpha_k = 1/(k+1)$ and that they have almost the same convergence rate when they use the same step sizes, as seen in Figures 2–5. It also revealed that all users' sequences generated by both algorithms converged to the same point. The details are omitted due to the lack of space.



(a) $D_1(k)$ vs. no. of iterations (b) $D_1(k)$ vs. elapsed time

Figure 4.: Behaviors of $D_1(k)$ for Algorithm 4.1 with $\alpha_k = 1/(k+1)$ and $10^{-3}/(k+1)$



(a) $F_1(k)$ vs. no. of iterations (b) $F_1(k)$ vs. elapsed time

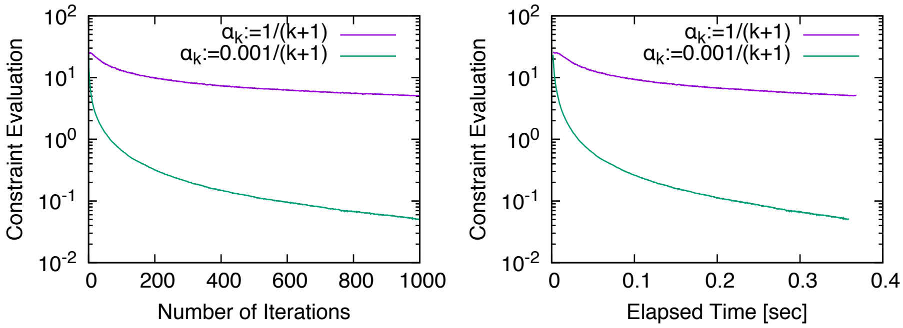
Figure 5.: Behaviors of $F_1(k)$ for Algorithm 4.1 with $\alpha_k = 1/(k+1)$ and $10^{-3}/(k+1)$

To illustrate the efficiency of Algorithms 3.1 and 4.1 for the whole network, let us observe the behaviors of $D(k)$ and $F(k)$ for all $k \geq 0$:

$$D(k) := \sum_{i=1}^{m} \left\| x_i(k) - \prod_{j=1}^{d} P_{X_i^j}(x_i(k)) \right\| \text{ and } F(k) := \sum_{i=1}^{m} f_i(x_i(k)).$$

If $(x_i(k))_{k\geq 0}$ $(i \in V)$ converge to the same point, the convergence of $(D(k))_{k\geq 0}$ to 0 implies that $(x_i(k))_{k\geq 0}$ $(i \in V)$ converges to a fixed point of $\prod_{j=1}^{d} P_{X_i^j}$ for all $i \in V$; i.e., to a point in $\bigcap_{i=1}^{m} \bigcap_{j=1}^{d} X_i^j = \bigcap_{i=1}^{m} X_i =: X$.

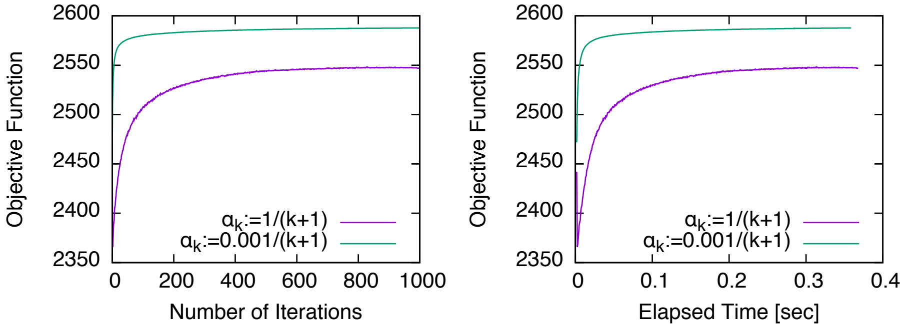
Figure 6 shows that $(x_i(k))_{k=0}^{1000}$ $(i \in V)$ generated by Algorithm 3.1 with $\alpha_k = 10^{-3}/(k+1)$ converged to a point in X faster than with $\alpha_k = 1/(k+1)$. Figure



(a) $D(k)$ vs. no. of iterations ($D(10^3) \approx 0.049868$ for $\alpha_k = 10^{-3}/(k+1)$)

(b) $D(k)$ vs. elapsed time

Figure 6.: Behaviors of $D(k)$ for Algorithm 3.1 with $\alpha_k = 1/(k+1)$ and $10^{-3}/(k+1)$



(a) $F(k)$ vs. no. of iterations ($F(10^3) \approx 2587.784266$ for $\alpha_k = 10^{-3}/(k+1)$)

(b) $F(k)$ vs. elapsed time

Figure 7.: Behaviors of $F(k)$ for Algorithm 3.1 with $\alpha_k = 1/(k+1)$ and $10^{-3}/(k+1)$

7 shows that $(F(k))_{k=0}^{1000}$ generated by Algorithm 3.1 with $\alpha_k = 10^{-3}/(k+1)$ was stable in the early stages.

Figures 8 and 9 illustrate the behaviors of $(D(k))_{k=0}^{1000}$ and $(F(k))_{k=0}^{1000}$ for Algorithm 4.1 with $\alpha_k = 1/(k+1), 10^{-3}/(k+1)$. Figure 9 shows that the value of $F(10^3)$ generated by Algorithm 4.1 with $\alpha_k = 10^{-3}/(k+1)$ was about 2587, which is almost the same as the value of $F(10^3)$ generated by Algorithm 3.1 with $\alpha_k = 10^{-3}/(k+1)$ (Figure 7). Although the value of $D(10^3)$ (≈ 0.049868) generated by Algorithm 3.1 with $\alpha_k = 10^{-3}/(k+1)$ was smaller than that (≈ 0.053781) generated by Algorithm 4.1 with $\alpha_k = 10^{-3}/(k+1)$, Figures 6–9 show that the behaviors for Algorithm 4.1 were almost the same as the ones for Algorithm 3.1.

Finally, the users are divided into 16 groups,

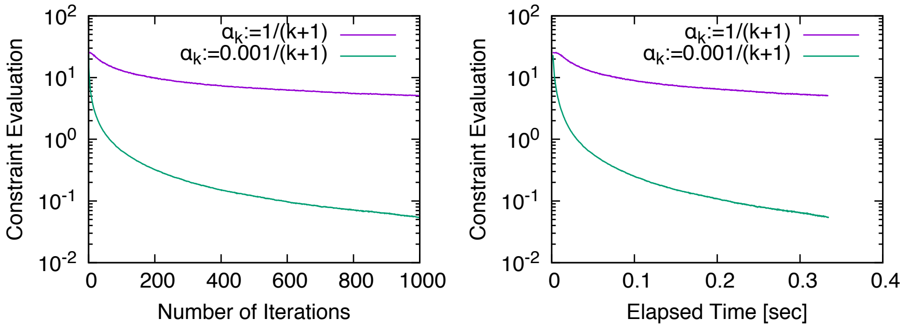
$$G_1 := \{2,3,4\}, \ G_2 := \{5,6,7\}, \ldots, \ G_{16} := \{47,48,1\},$$

and the values of the objective functions and performance measures for each group are compared.

$$F_{G_j} := \sum_{i \in G_j} F_i\left(10^3\right) \text{ and } D_{G_j} := \sum_{i \in G_j} D_i\left(10^3\right) \quad (j = 1,2,\ldots,16),$$

where $F_i(k)$ and $D_i(k)$ ($i \in V, k \geq 0$) are defined as in (28).

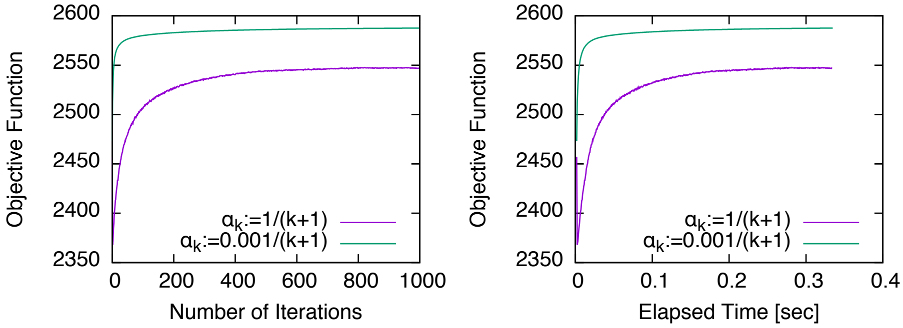
It can be seen from Table 1, which shows the values of F_{G_j} and D_{G_j} ($j = 1,2,\ldots,16$) for Algorithm 3.1 with $\alpha_k = 1/(k+1)$ and $10^{-3}/(k+1)$, that the D_{G_j}



(a) $D(k)$ vs. no. of iterations ($D(10^3) \approx$ 0.053781 when $\alpha_k = 10^{-3}/(k+1)$)

(b) $D(k)$ vs. elapsed time

Figure 8.: Behaviors of $D(k)$ for Algorithm 4.1 with $\alpha_k = 1/(k+1)$ and $10^{-3}/(k+1)$



(a) $F(k)$ vs. no. of iterations ($F(10^3) \approx$ 2587.630859 when $\alpha_k = 10^{-3}/(k+1)$)

(b) $F(k)$ vs. elapsed time

Figure 9.: Behaviors of $F(k)$ for Algorithm 4.1 with $\alpha_k = 1/(k+1)$ and $10^{-3}/(k+1)$

Table 1.: Values of $F_{G_j} := \sum_{i \in G_j} F_i(10^3)$ and $D_{G_j} := \sum_{i \in G_j} D_i(10^3)$ $(j = 1, 2, \ldots, 16)$ for Algorithm 3.1 with $\alpha_k = 1/(k+1)$ and $10^{-3}/(k+1)$

(a) $\alpha_k = 1/(k+1)$

Group	F_{G_j}	D_{G_j}
01	163.428093	0.325050
02	158.725168	0.301325
03	154.834519	0.340681
04	157.867759	0.325576
05	169.889020	0.342258
06	160.626418	0.304963
07	161.166806	0.328647
08	148.522159	0.342929
09	160.980531	0.270875
10	160.546361	0.356931
11	162.681703	0.326232
12	158.329823	0.361666
13	166.747981	0.329523
14	157.684287	0.253613
15	154.678365	0.326188
16	150.104832	0.292431

(b) $\alpha_k = 10^{-3}/(k+1)$

Group	F_{G_j}	D_{G_j}
01	166.805046	0.000154
02	161.267547	0.000816
03	157.564483	0.000780
04	160.620520	0.000016
05	172.271029	0.000024
06	162.896308	0.006480
07	163.472768	0.000000
08	151.511558	0.000712
09	163.481266	0.000000
10	163.301971	0.000316
11	164.912426	0.006555
12	160.847494	0.022754
13	169.300357	0.010475
14	160.186324	0.000000
15	156.894949	0.000785
16	152.450221	0.000000

Table 2.: Values of $F_{G_j} := \sum_{i \in G_j} F_i(10^3)$ and $D_{G_j} := \sum_{i \in G_j} D_i(10^3)$ ($j = 1, 2, \ldots, 16$) for Algorithm 4.1 with $\alpha_k = 1/(k+1)$ and $10^{-3}/(k+1)$

(a) $\alpha_k := 1/(k+1)$

Group	F_{G_j}	D_{G_j}
01	163.488124	0.319108
02	158.709368	0.299227
03	154.986973	0.332485
04	157.928141	0.324662
05	169.901684	0.321779
06	160.636388	0.308214
07	161.228299	0.314448
08	148.436659	0.362904
09	160.756450	0.266202
10	160.449748	0.368771
11	162.608469	0.340017
12	158.361547	0.347438
13	166.913645	0.321366
14	157.718005	0.240146
15	154.436379	0.348402
16	150.187445	0.286736

(b) $\alpha_k := 10^{-3}/(k+1)$

Group	F_{G_j}	D_{G_j}
01	166.798378	0.000058
02	161.254440	0.001169
03	157.543728	0.001018
04	160.614962	0.000035
05	172.259768	0.000000
06	162.899171	0.006912
07	163.458852	0.000000
08	151.501730	0.000813
09	163.477941	0.000000
10	163.295545	0.000309
11	164.885338	0.007128
12	160.815178	0.025368
13	169.319284	0.009938
14	160.189071	0.000000
15	156.886623	0.001033
16	152.430848	0.000000

values generated by Algorithm 3.1 with $\alpha_k = 10^{-3}/(k+1)$ were smaller than those generated with $\alpha_k = 1/(k+1)$. In particular, D_{G_4}, D_{G_7}, D_{G_9}, $D_{G_{14}}$, and $D_{G_{16}}$ were dramatically lower with $\alpha_k = 10^{-3}/(k+1)$. It can be seen from Table 2 that Algorithm 4.1 performed better with $\alpha_k = 10^{-3}/(k+1)$ than with $\alpha_k = 1/(k+1)$. This is because D_{G_5}, D_{G_7}, D_{G_9}, $D_{G_{14}}$, and $D_{G_{16}}$ were approximately zero with $\alpha_k = 10^{-3}/(k+1)$. Tables 1 and 2 show that the values of F_{G_j} generated by Algorithm 3.1 were almost the same as those generated by Algorithm 4.1. Accordingly, Algorithms 3.1 and 4.1 with the same step size have almost the same convergence rate and converge more quickly with $\alpha_k = 10^{-3}/(k+1)$ than with $\alpha_k = 1/(k+1)$, as seen in the figures.

The analyses in Subsections 3.2 and 4.2 and the results of the numerical evaluation indicate that the rate of convergence of Algorithm 3.1 is almost the same as that of Algorithm 4.1 when they have the same step size and that the algorithms are stable and converge quickly when they have smaller step sizes.

6. Conclusion and future work

The problem of minimizing the sum of all users' nonsmooth convex objective functions over the intersection of all users' closed convex constraint sets was discussed, and two distributed algorithms were presented for solving the problem. One algorithm uses each user's proximity operator and metric projection onto a set randomly selected from components of its constraint set while the other is obtained by replacing the proximity operator of the first algorithm with the subdifferential. Convergence analysis showed that, under certain assumptions, the sequences of all users generated by each of the two algorithms converge almost surely to the same solution to the problem. It also showed that the rates of convergence depend on the step size of the sequence and that it is desired to use small-step-size sequences so that the algorithms converge quickly. The results of numerical evaluation using a nonsmooth convex optimization problem support this analysis and demonstrate the effectiveness of the two algorithms.

The proposed algorithms work well when each user randomly sets one metric projection selected from many projections. Since nonexpansive mappings are generalizations of metric projections and thus have wider application, developing distributed random algorithms that work when one user randomly chooses one nonexpansive mapping at a time is a promising undertaking.

Acknowledgments I am sincerely grateful to the editor, Christiane Tammer, the anonymous associate editor, and the two anonymous reviewers for helping me improve the original manuscript. I also thank Kazuhiro Hishinuma for his input on the numerical evaluation.

References

[1] Bauschke, H.H., Combettes, P.L.: Convex Analysis and Monotone Operator Theory in Hilbert Spaces. Springer (2011)
[2] Beliakov, G., Li, G., Liu, S.: Parallel bucket sorting on graphics processing units based on convex optimization. Optimization **64**, 1033–1055 (2015)
[3] Bertsekas, D.P., Gallager, R.: Data Networks. Prentice Hall (1987)
[4] Bertsekas, D.P.: Incremental gradient, subgradient, and proximal methods for convex optimization: A survey. http://arxiv.org/pdf/1507.01030v1.pdf
[5] Bertsekas, D.P., Nedić, A., Ozdaglar, A.E.: Convex Analysis and Optimization. Athena Scientific (2003)
[6] Borkar, V.S.: Stochastic Approximation: A Dynamical Systems Viewpoint. Cambridge University Press (2008)
[7] Borwein, J.M., Lewis, A.S.: Convex Analysis and Nonlinear Optimization: Theory and Examples. Springer (2006)
[8] Brualdi, R.A., Ryser, H.J.: Combinatorial Matrix Theory. Cambridge University Press (1991)
[9] Bui, M.: On optimal management of resources in distributed networks. Optimization **21**, 785–796 (1990)
[10] Cai, X., Sha, D., Wong, C.K.: Time-Varying Network Optimization. Springer (2007)
[11] Casanova, H., Legrand, A., Robert, Y.: Parallel algorithms. Chapman and Hall/CRC (2009)
[12] Censor, Y., Zenios, S.: Parallel Optimization: Theory, Algorithms, and Applications. Oxford Univ Pr on Demand (1998)
[13] Combettes, P.L., Pesquet, J.C.: A Douglas-Rachford splitting approach to nonsmooth convex variational signal recovery. IEEE Journal of Selected Topics in Signal Processing **1**, 564–574 (2007)
[14] Goldsmith, A.: Wireless Communications. Cambridge University Press, Cambridge (2005)
[15] Iiduka, H.: Convergence analysis of iterative methods for nonsmooth convex optimization over fixed point sets of quasi-nonexpansive mappings. Mathematical Programming (to appear, DOI: 10.1007/s10107-015-0967-1)
[16] Iiduka, H.: Fixed point optimization algorithms for distributed optimization in networked systems. SIAM Journal on Optimization **23**, 1–26 (2013)
[17] Iiduka, H.: Multicast decentralized optimization algorithm for network resource allocation problems. Journal of Nonlinear and Convex Analysis **14**, 817–839 (2013)
[18] Iiduka, H.: Acceleration method for convex optimization over the fixed point set of a nonexpansive mapping. Mathematical Programming **149**, 131–165 (2015)
[19] Iiduka, H.: Proximal point algorithms for nonsmooth convex optimization with fixed point constraints. European Journal of Operational Research **253**, 503–513 (2016)
[20] Iiduka, H.: Almost sure convergence of random projected proximal and

subgradient algorithms for distributed nonsmooth convex optimization. http://arxiv.org/pdf/1510.07107.pdf (2015)

[21] Iiduka, H., Hishinuma, K.: Acceleration method combining broadcast and incremental distributed optimization algorithms. SIAM Journal on Optimization **24**, 1840–1863 (2014)

[22] Iiduka, H., Yamada, I.: A use of conjugate gradient direction for the convex optimization problem over the fixed point set of a nonexpansive mapping. SIAM Journal on Optimization **19**, 1881–1893 (2009)

[23] Kennington, J., Olinick, E., Rajan, D. (eds.): Wireless Network Design. Springer (2011)

[24] Lee, S., Nedić, A.: Distributed random projection algorithm for convex optimization. IEEE Journal of Selected Topics in Signal Processing **7**, 221–229 (2013)

[25] Maillé, P., Toka, L.: Managing a peer-to-peer data storage system in a selfish society. IEEE Journal on Selected Areas in Communications **26**(7), 1295–1301 (2008)

[26] Nedić, A.: Random algorithms for convex minimization problems. Mathematical Programming **129**, 225–253 (2011)

[27] Nedić, A., Bertsekas, D.: Incremental subgradient methods for nondifferentiable optimization. SIAM Journal on Optimization **12**, 109–138 (2001)

[28] Nedić, A., Ozdaglar, A.: Cooperative distributed multi-agent optimization. Convex Optimization in Signal Processing and Communications, pp. 340–386 (2010)

[29] Pilanci, M., Wainwright, M.J.: Randomized sketches of convex programs with sharp guarantees. http://arxiv.org/abs/1404.7203

[30] Pop, P.C.: Generalized Network Design Problems. Walter de Gruyter GmbH (2012)

[31] Prabhakar, B., Kulkarni, A.A.: Dimensionality reduction of affine variational inequalities using random projections. http://arxiv.org/abs/1408.4551

[32] Ram, S.S., Nedić, A., Veeravalli, V.V.: A new class of distributed optimization algorithms: Application to regression of distributed data. Optimization Methods and Software **27**, 71–88 (2012)

[33] Srikant, R.: The Mathematics of Internet Congestion Control. Birkhauser (2004)

[34] van der Vaart, A.W.: Asymptotic Statistics. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press (2007)

[35] Vempala, S.S.: The Random Projection Method. American Mathematical Society (2004)

[36] Wang, M., Bertsekas, D.P.: Stochastic first-order methods with random constraint projection. SIAM Journal on Optimization **26**, 681–717 (2016)

[37] Xiao, L., Boyd, S.: Fast linear iterations for distributed averaging. Systems and Control Letters **53**, 65–78 (2004)

[38] Yamada, I., Yukawa, M., Yamagishi, M.: Minimizing the Moreau envelope of nonsmooth convex functions over the fixed point set of certain quasi-nonexpansive mappings. In: H.H. Bauschke, R.S. Burachik, P.L. Combettes, V. Elser, D.R. Luke, H. Wolkowicz (eds.) Fixed-Point Algorithms for Inverse Problems in Science and Engineering, pp. 345–390. Springer (2011)

[39] Yin, X.C., Huang, K., Hao, H.W., Iqbal, K., Wang, Z.B.: A novel classifier ensemble method with sparsity and diversity. Neurocomputing **134**, 214–221 (2014)

[40] Yin, X.C., Huang, K., Yang, C., Hao, H.W.: Convex ensemble learning with sparsity and diversity. Information Fusion **20**, 49–58 (2014)